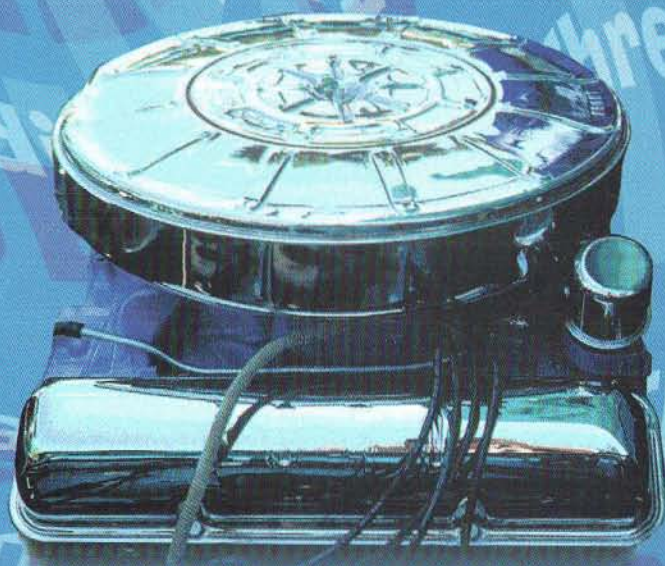


MacTech®

DataBase Engines

NeoAccess 5.0 Revealed

dtF/SQL: The Little Engine that Could



**OpenBase - A Solid Database
Framework for Rhapsody**

A hand holding a globe with Java code floating around it. The code includes comments like "Sets the true value of the boolean", "if true, the", "if false, the", and "if the specified", as well as method signatures like "public void setAllowSorting(boolean oldvalue)".

For Macintosh

1-800-971-1582 MT121

*All features included in the Database Development Edition **Features as of October 15, 1997

Symantec and the Symantec logo are U.S. registered trademarks and Symantec Visual Café is a trademark of Symantec Corporation. Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems in the U.S. and other countries. Apple, Macintosh and Power Macintosh are registered trademarks of Apple Computer, Inc. All other brands and products are trademarks of their respective holder/s. ©1997 Symantec Corporation. All rights reserved. In Canada, call 1-800-365-8641. In Australia, call 02-9850-1000. In Europe, call 31-71-535-3111.

 Mac OS

With Diamonds
This Perfect,
The Only Thing
We Cut
Are The Prices.

Mitsubishi's full line of award-winning DIAMONDTRON™ aperture grille monitors have a proven reputation for quality, reliability, and value. And now you can get all this at prices lower than ever before. Plus, you'll have added confidence knowing that all Mitsubishi Diamond Plus® and Diamond Pro® Series monitors are backed by a full 3-year warranty, combined with service and support programs that are already the best in the business. With the new lower pricing, everyone can experience the DIAMONDTRON™ difference. Get all the details by contacting Mitsubishi Electronics today.



USA: 1-800-843-2515 • Canada: 1-800-387-9630 • Qwikfax: 1-800-937-2094
World Wide Web: www.mitsubishi-display.com

Diamond Pro® Series

DIAMONDTRON solutions for graphics professionals

New! Only \$1799
Diamond Pro 1000

- 21"/19.7" DVI*
- 1800 x 1440/76 Hz
- 30-115 kHz auto-scan range

Reduced to \$1399
Diamond Pro 91TXM

- 21"/19.7" DVI*
- 1600 x 1200/75 Hz
- 30-95 kHz auto-scan range

Reduced to \$849
Diamond Pro 700

- 17"/16.0" DVI*
- 1600 x 1200/75 Hz
- 30-95 kHz auto-scan range

Diamond Plus® Series

DIAMONDTRON displays for people who mean business

New! Only \$1499
Diamond Plus 100e

- 21"/19.7" DVI*
- 1600 x 1200/85 Hz
- 30-108 kHz auto-scan range
- USB-ready

Reduced to \$679
Diamond Pro 87TXM

- 17"/16.0" DVI*
- 1600 x 1200/66 Hz
- 30-86 kHz auto-scan range

New! Only \$599
Diamond Plus 70

- 17"/16.0" DVI*
- 1280 x 1024/65 Hz
- 30-70 kHz auto-scan range



MITSUBISHI
DISPLAY PRODUCTS
Innovation On Display_{sm}

©1998 Mitsubishi Electronics America, Inc. DIAMONDTRON is a trademark of Mitsubishi Electric Corporation. Diamond Pro and Diamond Plus are registered trademarks of Mitsubishi Electronics America, Inc. Microsoft, Windows and the Windows logo are registered trademarks of Microsoft Corporation. Mac and the Mac OS logo are trademarks of Apple Computer, Inc. Monitor prices and specifications subject to change without notice. *DVI (Diagonal Viewable Image)

How To Communicate With Us

In this electronic age, the art of communication has become both easier and more complicated. Is it any surprise that we prefer **e-mail**?

If you have any questions, feel free to call us at 805/494-9797 or fax us at 805/494-9798.

If you would like a subscription or need customer service, feel free to contact Developer Depot Customer Service at 800-MACDEV-1

DEPARTMENTS

Orders, Circulation, & Customer Service

Press Releases

Ad Sales

Editorial

Programmer's Challenge

Online Support

Accounting

Marketing

General

Web Site (articles, info, URLs and more...)

E-Mail/URL

cust_service@mactech.com

press_releases@mactech.com

ad_sales@mactech.com

editorial@mactech.com

prog_challenge@mactech.com

online@mactech.com

accounting@mactech.com

marketing@mactech.com

info@mactech.com

http://www.mactech.com

MACTECH MAGAZINE

MacTech Magazine is grateful to the following individuals who contribute on a regular basis. We encourage others to share the technology.

We are dedicated to the distribution of useful programming information without regard to Apple's developer status.

*For information on submitting articles, ask us for our **writer's kit** which includes the terms and conditions upon which we publish articles.*

Editorial Board of Advisors

Scott T Boyd, Jordan J. Mattson, Jim Straus, and Jon Wiederspan

Editors

Publisher • Neil Ticktin

Editor Emeritus • Eric Gundrum

Editor • Nick "nick.e" DeMello

Managing Editor • Jessica Courtney

Online Support • Kevin Avila

Contributing Editors

- Gary Little, Apple Computer, Inc.
- Tantek Çelik, Microsoft Corporation
- Carl de Cordova
- Jeff Ganyard
- Rich Siegel, Bare Bones Software, Inc.
- Steve Kiene, Mindvision
- Richard Clark
- Jim Gochee, Connectix
- Ed Ringel
- Michael Rutman

Regular Columnists

Getting Started • Dave Mark

Programmer's Challenge • Bob Boonstra

From the Factory Floor • Dave Mark, Metrowerks

Tips & Tidbits • Steve Sisak

MacTech Online • Jeff Clites

XPLAIN CORPORATION

Chief Executive Officer • Neil Ticktin

Chief Operating Officer • Andrea J. Sniderman

Controller • Heather Principe

Advertising Executive • Fran Stern

MarComm Manager • Susan M. Whitney

Events Manager • Susan M. Worley

Network Administrator • Chris Barrus

Customer Relations • Lee Ann Pham,
Susan Pomrantz, Lenell Solomon

Accounting • Paula Garrety, Jan Webber

Art Direction/Production • InfoGraphix

Board of Advisors • Steven Geller, Blake Park,
and Alan Carsrud



This publication is
printed on paper with
recycled content.

All contents are Copyright 1984-1998 by Xplain Corporation. All rights reserved. MacTech is a registered trademark and MacDev-1, THINK Reference, Developer Depot, Sprocket, JavaTech, WebTech, BeTech, and the MacTutorMan are trademarks of Xplain Corporation. develop is a trademark of Apple Computer, Inc. Other trademarks and copyrights appearing in this printing or software remain the property of their respective holders. Xplain Corporation does not assume any liability for errors or omissions by any of the advertisers, in advertising content, editorial, or other content found herein. Opinions or expressions stated herein are not necessarily those of the publisher and therefore, publisher assumes no liability in regards to said statements.

MacTech Magazine (ISSN: 1067-8360 / USPS: 010-227) is published monthly by Xplain Corporation, 850-P Hampshire Road, Westlake Village, CA 91361-2800. Voice: 805/494-9797, FAX: 805/494-9798. Domestic subscription rates are \$47.00 per year. Canadian subscriptions are \$59.00 per year. All other international subscriptions are \$97.00 per year. Domestic source code disk subscriptions are \$77 per year. All international disk subscriptions are \$97.00 a year. Please remit in U.S. funds only. Periodical postage is paid at Thousand Oaks, CA and at additional mailing office.

POSTMASTER: Send address changes to **MacTech Magazine**, P.O. Box 5200, Westlake Village, CA 91359-5200.

Contents

April 1998 • Volume 14, No.4

For Macintosh
Programmers & Developers

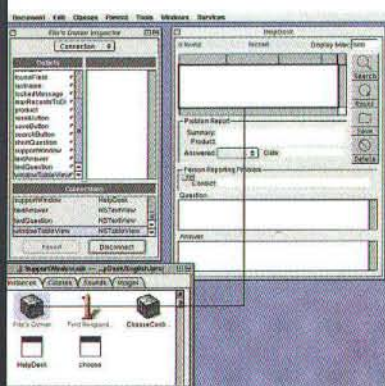
MacTech
M A G A Z I N E

Feature Articles

- 22 TOOLS OF THE TRADE**
NeoAccess 5.0 Revealed
Taking a look at this high-performance object database management system
by Michael Marchetti
- 47 dtF/SQL — The Little Engine That Could**
sLab makes their powerful SQL database engine available for anyone to use, at little or no cost
by William A. Gilbert, Ph.D.

Columns

- 4 VIEWPOINT**
by Ed Ringel
- GETTING STARTED**
- 6 Having a Dialog with Your Mac**
by Dave Mark
- FROM THE FACTORY FLOOR**
- 36 Neil Ticktin, MacTech Publisher**
by Dave Mark
- PROGRAMMER'S CHALLENGE**
- 58 Mancala**
by Bob Boonstra
- 70 MACTECH ONLINE**
by Jeff Clites



NeoAccess 5.0 page 30

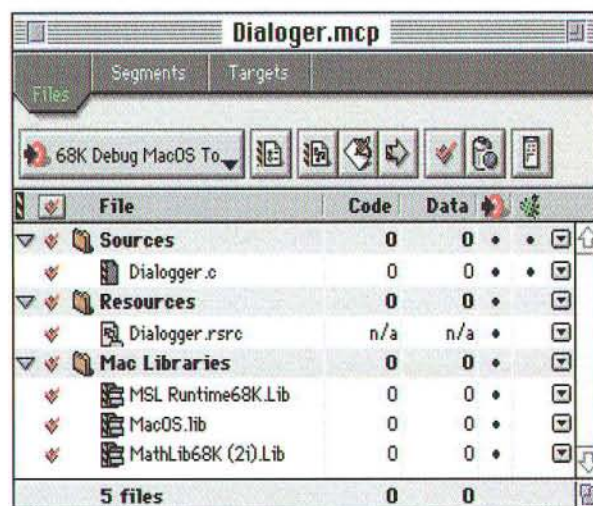
- 50 Jovis, a Database Engine in a Tiny Package**
A robust database engine for HyperCard and other XCMD compatible environments
by Edward Ringel

- 53 Macworld Developer Tools Roundup**
What's new on the tools for developers
by Michael Rutman

- 30 RHAPSODY**
OpenBase
A solid database framework for Rhapsody
by Gene Backlin

- 40 EMERGING TECHNOLOGIES**
Using Apple Information Access Toolkit
Apple's new indexing technology provides powerful ways to search and retrieve your data, regardless of how it's stored
by Mark Holtz

- 56 JAVA WORKSHOP**
Java Serialization
Adding object persistence to Java applications
by Andrew Downs



Getting Started

page 6

Reader Resources

- 69 Tips & Tidbits**
- 72 NewsBits**
- 76 The Classifieds**
- 78 Advertiser & Product Index**

by Ed Ringel

MACTECH MAGAZINE'S DATABASE PROJECT

With the focus of this issue on database engines, I've asked Ed Ringel, MacTech's Contributing Editor for product reviews, to tell us a little bit about why he thinks database technologies are crucial to all Macintosh developers. Ed has been working for many months to bring us information about many of the database engines available for use in Macintosh programs. The results of some of his work appear in this issue; he also is responsible for bringing us several database articles over the past year and a half, and yet more will appear in future issues. I hope you enjoy his guest editorial. — Eric Gundrum, Editor Emeritus

If you riffle through the last year's or so issues of MacTech, you will note several database related articles, such as those about PowerBuilder for Macintosh and B-Tree Helper. This issue is a continuation of our attempt to address one of the basic issues of computer programming and software development, the management of large amounts of data.

Certainly, many off the shelf products do not deal with data that is managed in a database — graphics and word processing programs come to mind almost immediately. However, such diverse functions as investing our mutual funds and querying our bank accounts, managing our health care, the management of corporate America, and our right to vote depend on the ability to manipulate formatted information in a predictable, easy, and useful manner. Those of you who do contract programming almost certainly have undertaken a database or information management product at some time or another. Sadly, the Mac OS often has not been chosen as the platform for management of this kind of information. This is partially because of the artificial ascendancy of the Wintel platform, but it also is partially because of the perception of many IS professionals that the Mac has inadequate tools for the task. We want to put the latter misperception, at least, to rest.

In this issue we present articles on two high end products dtF and NeoAccess, for your evaluation, and an additional article on a rather interesting find, Jovis. dtF and NeoAccess are powerful databases with inherent cross platform capabilities. Jovis, although inherently not as versatile as the other two because of its confinement to the XCMD (and thus the Mac OS) world, nonetheless does what it does very well. All three of these products can do a fine job of managing complex information for many users.

All three of these products also support the developer in the development of static knowledge bases with the capability to handle large, arbitrary blocks of binary data. Although a departure from "traditional" information management, this is a growth area for the use of database engines. This should be an area of particular interest to the Mac developer. The Mac's beautiful graphics, excellent printing, and ease of programming multimedia make the platform a natural for static knowledge bases.

What am I talking about? As I browse the catalogues and particularly browse the store shelves, much of the software sold at the retail level simply is not a new way of manipulating data. Rather, the purchaser is buying the information on the CD, not the program that presents it. Sadly, it is very hard to come up with a new program that genuinely is a better mousetrap, and many talented developers spend much time and treasure trying to write the next PageMaker or Excel. Face it guys, for most of us, that ain't in the cards. Additionally, the reality is that the programs out there that do most computational tasks are pretty good; the competition is pretty stiff. Reflecting this fact, and reflecting the public's hunger for consumable stuff, concentrating on information rather than software, may be a better tack for many developers. Many interactive games (short of arcade games) use a relatively simple shell that really is a multimedia display with decision points and programmed responses. Teaching CD's, information CD's, and my movies CD from my local video store don't sell the information player, they sell the information.

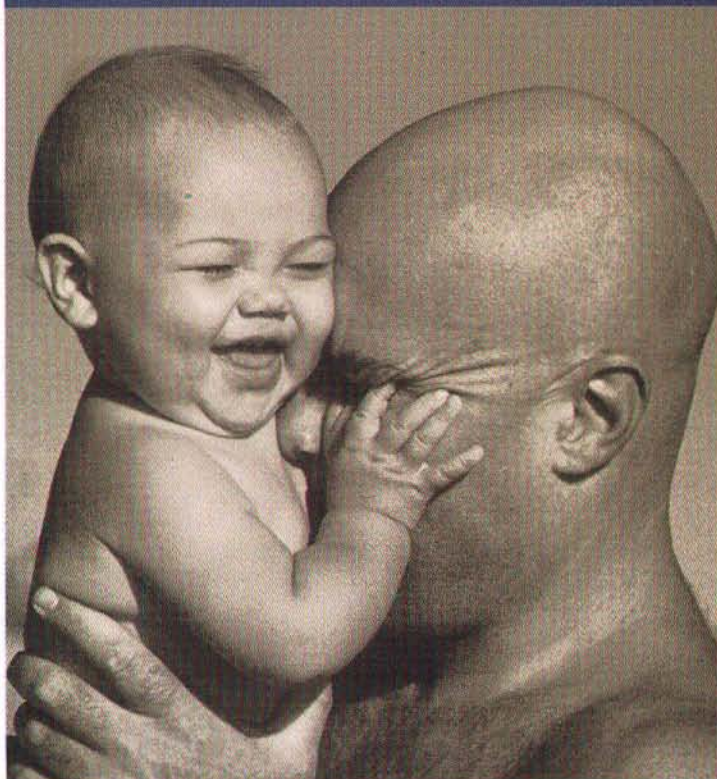
Arguably then, content development is pretty darn important. Good quality, easy to use tools such as those we've reviewed here give the developer breathing room to think about content, structure, and esthetic appeal of the presentation. Certainly, we will continue to want high-quality code programmers to flower and prosper, but many of us can create significant works with our good understanding of communication and esthetics and by collaborating with our colleagues who are content experts. This may be more important to professional success than shaving another microsecond from a rendering routine. Think about it.

Our reviews of this important area are not complete. We invite vendors and readers to let us know about their favorite engines, and we'll try to evaluate them. Be careful, though, you might get tapped to be the author of the review! **MT**

Ed Ringel is Contributing Editor for product reviews for MacTech Magazine. When he's not working at the computer or enjoying the Maine lakes and woods, he's a respiratory and critical care physician in Waterville, Maine. He can be reached at eringel@mint.net

MacHASP®

Protects Your Software



MacHASP – The Professional Software Protection System*

- Fully licensed ADB device
- Unequalled security
- Unparalleled flexibility
- Genuine ease-of-use and transparency
- Supports System 6.05 and up, including System 8.1
- ISO 9002 quality and reliability



HASP Protects More

See Us At Comdex Spring '98 – Booth No. 1280

North America Aladdin Knowledge Systems Inc. 800 223 4277, 212 564 5678, Fax: 212 564 5577, Email: hasp.sales@us.aks.com
 Int'l Office Aladdin Knowledge Systems Ltd. +972 3 636-2222, Fax: +972 3 537-5796, Email: hasp.sales@aks.com
 Germany Aladdin Knowledge Systems GmbH & Co. KG +49 89 89 42 21-0, Fax: +49 89 89 42 21-40, Email: info@aladdin.de
 UK Aladdin Knowledge Systems UK Ltd. +44 1753 622266, Fax: +44 1753 622262, Email: sales@aladdin.co.uk
 Japan Aladdin Japan Co., Ltd. +81 426 60-7191, Fax: +81 426 60-7194, Email: sales@aladdin.co.jp
 France Aladdin France SA +33 1 41-57-70-30, Fax: +33 1 41-57-70-39, Email: info@aladdin.fr
 Benelux Aladdin Software Security Benelux B.V. +31 24 648 8444, Fax: +31 24 645 1981, Email: sales@aladdin.nl
 Russia Aladdin Software Security R.D. Ltd. +7 095 923-0588, Fax: +7 095 928-6781, Email: aladdin@aladdin.msk.ru

Your software is your baby – and you want to look after it. You created it, you developed it, you saw it right through to the moment it was ready for market. Now protect it. 50% of business software is stolen; \$11 billion of developers' income is lost to piracy.* Is your software a statistic?

All over the world, more developers are protecting against piracy. They're protecting more products, on more platforms, with more security – and selling more as a result. And more of these developers are protecting with MacHASP.

To see why 25,000 developers worldwide protect with Aladdin, call and order your MacHASP Developer's Kit now!



1-800-223-4277
www.aks.com

ALADDIN®

The Professional's Choice

■ Australia Contact 03 98985685 ■ China (Beijing) Telex 010 62567389 (Hong Kong) Hastings 0755 2328741 ■ Czech Atlas 02 768085 ■ Denmark Berendsen 039 577318 ■ Egypt Zeitelstein 02 3804632 ■ Finland ID-Systems 09 8703520
 ■ Greece Unibrain 01 6756320 ■ India Solution 011 2148254 ■ Italy Partner Data 02 26147380 ■ Korea Dae-A 02 8484481 ■ Mexico Soft 091 80055263 ■ Poland Systherm 061 480273 ■ Portugal Futarmatica 01 4116269 ■ Romania Ro Interactive 064 140383
 ■ Singapore ITR 065 5666788 ■ South Africa D I e Roux 011 8864704 ■ Spain PC Hardware 03 4493193 ■ Sweden Kordab 455 307 300 ■ Switzerland Opag 061 7169222 ■ Taiwan Tecco 02 5559676 ■ Turkey Mikrohela 0312 4670635 ■ Yugoslavia Asys 021 623920

© Aladdin Knowledge Systems Ltd. 1995-1998. MacHASP and (T) HASP is a registered trademark of Aladdin Knowledge Systems Ltd. Mac & the Mac OS logo are trademarks of Apple Computer, Inc., used under license. All other product names are trademarks of their respective owners.

* 1997 BSA/SPA Study

by Dave Mark

Having a Dialog with Your Mac

Programming the Dialog Manager

In last month's column, we explored the inner workings of the Menu Manager. This month, we'll build on that experience and create a program with menus, windows, and, for the first time, dialogs and alerts. This program is the biggest one we've tackled so far, so this month's column is a bit longer than usual.

CREATING THE DIALOGGER RESOURCES

We'll start off just as we did in last month's column, by creating our MBAR and MENU resources. Create a folder inside your Development folder named "Dialogger Folder." Next, launch ResEdit and create a new file called Dialogger.rsrc inside this new folder.

Now select **Create New Resource** from the **Resource** menu and create a new MBAR resource according to the specifications in **Figure 1**. As you can see, our program will feature three different menus.

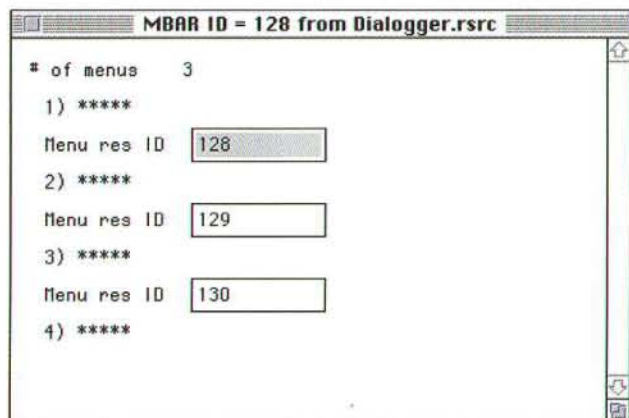


Figure 1. Specifications for Dialogger's MBAR resource.

Close the MBAR and MBAR picker windows and again select **Create New Resource** from the **Resource** menu. Create a MENU resource according to the specifications in **Figure 2**. This MENU represents the Apple menu. Be sure that the MENU's resource id is 128.

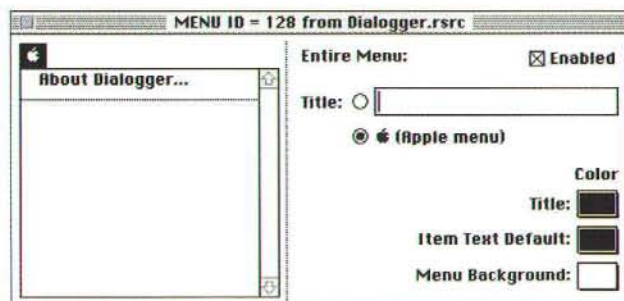


Figure 2. Specifications for the Apple MENU resource.

Now create another MENU resource according to the specifications shown in **Figure 3**. This MENU represents the File menu. Be sure the MENU's resource id is set to 129.

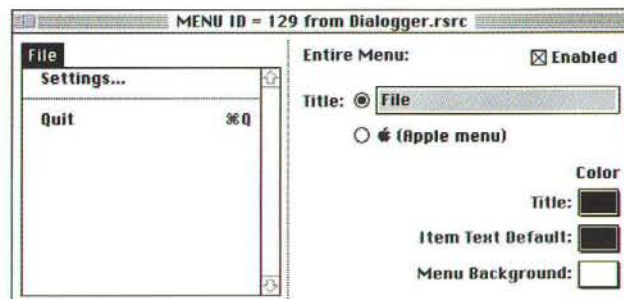


Figure 3. Specifications for the File MENU resource.

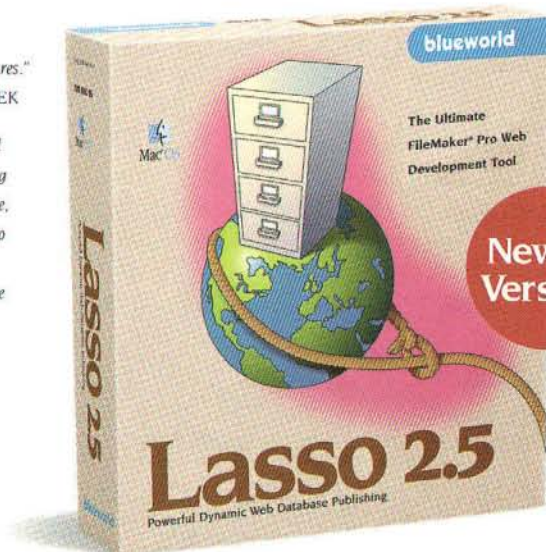
Now create one final MENU resource according to the specifications shown in **Figure 4**. This MENU represents the Edit menu. Be sure this MENU's resource id is set to 130.

Want to suggest an article for the magazine? Send your suggestion to
<mailto:editorial@mactech.com>

The Sharpest Tool On The Cutting Edge Of Web Database Design

"Lasso's strengths are its extensive built-in commerce and e-mail features."
MacWEEK

"With excellent support for advanced FileMaker Pro features, outstanding formatting control, fast performance, and thorough documentation, Lasso is a database-publishing demon."
Macworld Magazine



"Lasso has proven to be a great resource for the Mac OS Software & Hardware Guide. We are very impressed with the speed of the searches in a database of this size. Your company support has been great too!"

Gayle Westbrook
Software & Hardware Guide
Product Manager at
Apple Computer.

"Lasso simply blows Tango out of the water"
Paul Marty
University of Illinois

The Ultimate FileMaker® Pro Web Development Tool

Why do scores of high-profile websites rely on Lasso? Why is a portion of Lasso technology embedded in FileMaker Pro 4.0's Web Companion? With a feature set described as incredible, find out for yourself why Lasso 2.5 is the sharpest tool for creating online stores, discussion groups, response forms and countless other robust dynamic Web applications.

Over 50 new tags including math, string and variable allow unparalleled data handling capability. Combined with an SSL server, Lasso allows for secure data transactions. Create embedded operations with the powerful inline command. Instantly upgrade FileMaker 4.0 CDML tags to LDML with the Lasso Tag Converter. What's more, Lasso 2.5 Server edition supports multi-homing and offers superior performance over other Web servers.

Find out today why Lasso is the award-winning tool which sets the standard for Mac OS dynamic Web database publishing. To order or for a reseller near you, please call 425.646.0288. Download your free test drive version today at

<http://www.blueworld.com/lasso/download/>.

blueworld

bringing business to the internet™

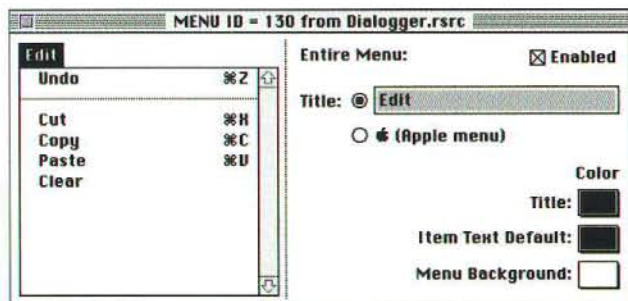


Figure 4. Specifications for the Edit MENU resource.

OK. Close up your MENU and MENU picker windows. This next part might be a little tricky. You'll need three PICT images, one of a dog, one of an elephant, and one of a squirrel. If you don't have any of these handy, drop into your favorite graphics program and do the best you can. If you're really stuck, just create pictures with the words *Afghan*, *Elephant*, and *Squirrel* in them. Paste all three pictures into your scrapbook.

Back in ResEdit, **Paste** the pictures into three PICT resources. Use **Figure 5** as your guide. Be sure your afghan is PICT 128, your elephant is PICT 129, and that your squirrel is PICT 130.

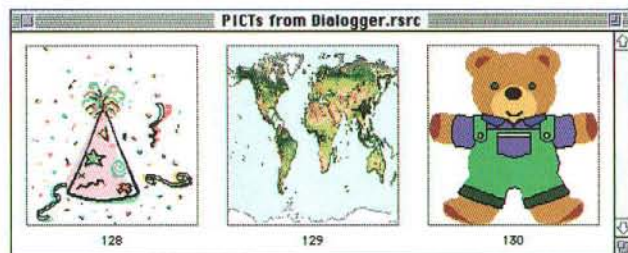


Figure 5. Three PICT resources. Note which resource id goes with which PICT.

Actually, you can use any PICT images that strike your fancy. Once you understand what this program does and how it works, feel free to customize it to your own desires.

Close the PICT picker window. Once again, select **Create New Resource** from the **Resource** menu. This time, create a DLOG resource. A miniature version of your desktop will appear with a window somewhere in the middle. Depending on the size of your monitor, your DLOG window may appear somewhat scaled. When the DLOG editor appears, customize it according to the specifications in **Figure 6**.

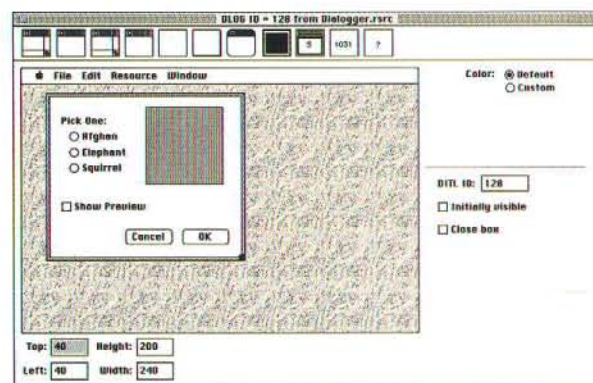


Figure 6. Specifications for the DLOG resource.

Be sure that you click the dialog window type (8th from the left) at the top of the editing window. Also, be sure you uncheck the **Initially visible** and **Close box** check boxes on the right side of the window. This resource controls the appearance of a dialog box's window. Now we'll create a resource that defines the items that appear in this DLOG.

Double-click the DLOG window (not the editing window, but the window that appears in the middle of the editing window, on the mini-desktop) and a DITL editing window will appear. At the same time, a DITL item palette will appear (**Figure 7**). This palette contains a list of all the items you can add to your dialog.

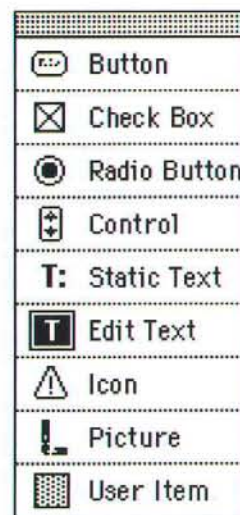


Figure 7. The DITL item palette.

When you start constructing a DITL (Dialog Item List), you always start with the **OK** and **Cancel** buttons. The **OK** button is always item number one, and the **Cancel** button (if it exists) is always item number two. As you'll learn, these item numbers have special significance to the Dialog Manager.

Click the **Button** palette, dragging to the left, on to your DITL window. The outline of a button will appear (**Figure 8**).

GEEK OUT

at...

***"The MUST attend
conference if you are
serious about Mac
programming...or
want to be!"***



With support from Apple® and MacTech® Magazine, MacHack is "THE" conference on Macintosh programming and development. MacHack gives you the opportunity to meet and talk to Apple engineers who actually work on the technologies you will be using tomorrow. Network with Macintosh developers from hundreds of companies and discuss programming techniques and issues you are concerned about. Unlike conferences where you are being talked "at", at MacHack you will be conversing with other programmers about real topics that affect you on a daily basis. This conference, by definition, is technical!

**SPACE IS LIMITED
REGISTER TODAY!**

MacHack 1998
The 13th Annual Conference
for Leading Edge Developers
June 18-21, 1998
Dearborn, Michigan
Keynote Speaker:
Chris Espinosa, "Apple Employee #8"

You are guaranteed exposure to some of the most technically advanced people in the industry. The conference is kept to a small size to encourage interaction. Events are created to foster relationship building, sharpen your coding skills, and educate you on a variety of topics – no matter what your skill level. In short, you'll mingle, network and learn at MacHack 1998!

But that's not all. Developers at MacHack are at the bleeding edge of new and emerging technologies. Developers for Rhapsody, Java, Pilot, BeOS and WinCE wander around, just waiting to share their knowledge with all who are willing to ask.

Registration Fees until April 15, 1998:

Attendee: \$425 Speaker: \$325

After April 15th:

Attendee: \$525 Speaker: \$420

If you would like to be a speaker, please contact Expotech at the address below.

**For more information or to register
contact Expotech at:**

1264 Bedford Rd.

Grosse Pointe Park, MI 48230-1116

Phone: 313-882-1824

FAX: 313-882-6942

E-MAIL: expotech@aol.com

WEB: www.machack.com

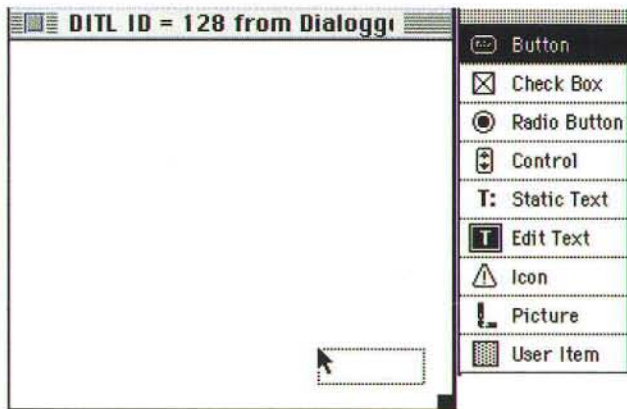


Figure 8. Dragging a button from the palette window.

When you release the mouse button, a new button item will appear. Double click this button item and a specifications window will appear. Fill it in according to the specs shown in **Figure 9**. Notice that the **Enabled** check box is checked. This tells the Dialog Manager to respond to this item when it is clicked on in a dialog box. If the item were not enabled, clicks in it would be ignored.

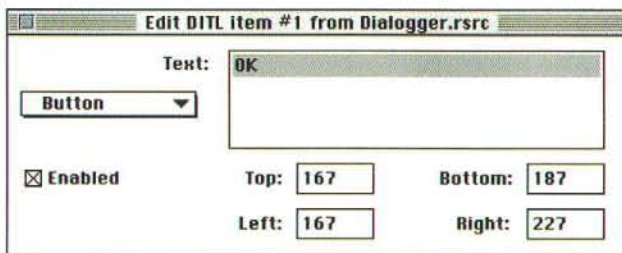


Figure 9. Specifications for DITL item #1.

Close the spec window and drag a second button from the palette. Double-click it and enter the specs shown in **Figure 10**. Close the specs window.

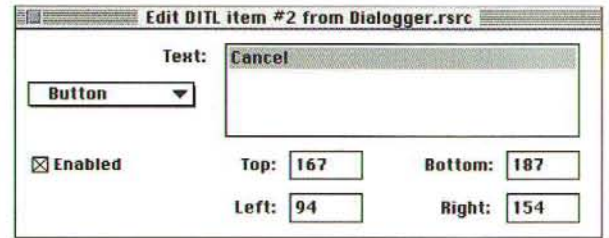


Figure 10. Specifications for DITL item #2.

This time, drag a Radio Button from the palette. Double-click it and make it reflect the specs shown in **Figure 11**. Close the specs window.

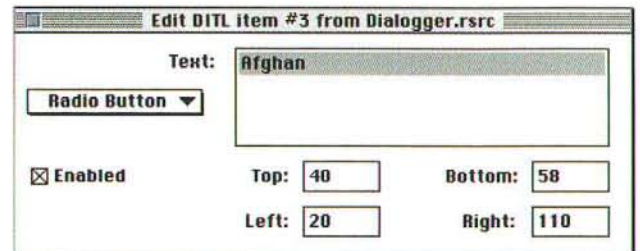


Figure 11. Specifications for DITL item #3.

Drag a second Radio Button from the palette and make it reflect the specs shown in **Figure 12**.

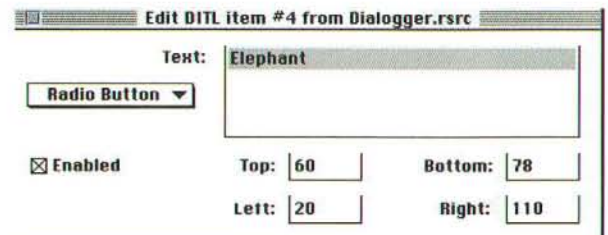


Figure 12. Specifications for DITL item #4.

Drag a third Radio Button from the palette and make it reflect the specs shown in **Figure 13**.

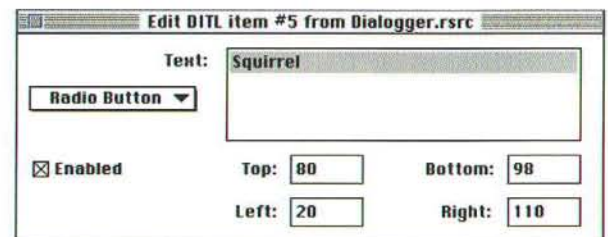


Figure 13. Specifications for DITL item #5.

Want to subscribe to MacTech Magazine? Contact
<mailto:orders@devdepot.com>,
call 800-MACDEV-1 or
805-494-9797, or
fax 805-494-9798

These days, many AOL users are finding themselves S.O.L.

AOL: America Off Line?

Even though AOL is by far the largest online service, it now costs more and continues to be among the least accessible. The fact is, many subscribers still wait to get on AOL far longer and more often than with other providers.

Access, of course, is everything. And if you're looking to AOL to provide it, you're *?!# out of luck. With EarthLink, you'll find that we're up 99.9% of the time. You won't be burdened with the annoying busy signals and disconnects AOL is famous for (nor their high prices: we're still \$19.95 a month). And because we have more local access numbers than any other Internet Service Provider, you can also get on the Net more easily. Speaking of easily, that's how you can get in touch with us. Our phone number is cleverly listed below. Talk to you soon.

PC MAGAZINE ISP RATINGS (9/97)

	EarthLink	AOL
Call Rate Success	Good	Poor
Average throughput (download speed)	Excellent	Poor
Web throughput Success	Excellent	Good
Overall Performance	Good	Poor

(888)QUIT-AOL, ext. 3926

EarthLink **Still \$19.95**
PER MONTH



It's your Internet.™

Next, drag a Static Text item from the palette and make it reflect the specs in **Figure 14**. Notice that the **Enabled** check box is unchecked. Normally, user clicks in static text are ignored.



Figure 14. Specifications for DITL item #6.

Next, drag a Check Box item from the palette and make it reflect the specs in **Figure 15**.

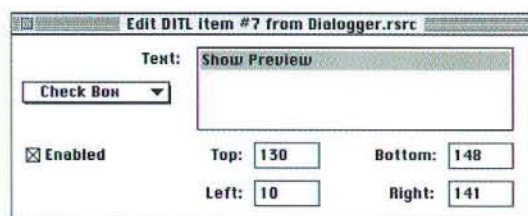


Figure 15. Specifications for DITL item #7.

Finally, drag a User Item from the palette and make it reflect the specs in **Figure 16**. As was the case with Static Text, notice that the **Enabled** check box is unchecked. A User Item just acts as a marking rectangle. We'll use it as a guide for drawing a picture, later in the program. There may be times when you want your User Items enabled. For now leave it as is.

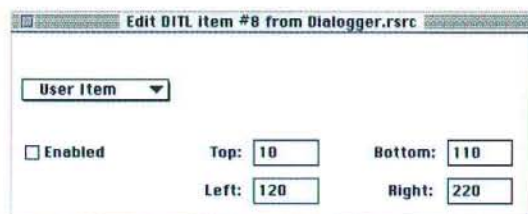


Figure 16. Specifications for DITL item #8.

When you close your last item spec window, your DITL editing window should look like the one shown in **Figure 17**. If it doesn't, go back and check out your specs. If you need to make changes, you can use the tools in the DITL menu, such as Show Item Numbers, Set Item Number..., and Renumber Items....

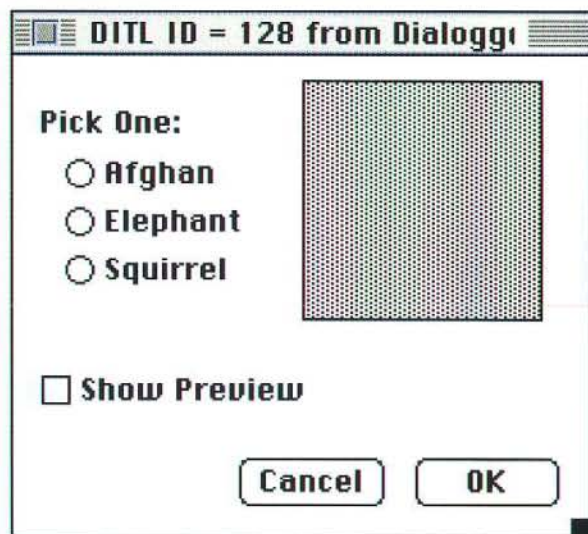


Figure 17. The completed DITL resource.

OK, we're almost done. Close the DITL and DLOG windows till you're back down to your main window. You might want to save at this point. I'll wait...

Select **Create New Resource** from the **Resource** menu and create an ALRT resource. The ALRT editor looks just like the DLOG editor. It should. Alerts are basically simplified dialogs, easier to use and easier to build. (Top: 40, Left: 40, Bottom: 145, Right: 350)

Select **Get Resource Info** from the **Resource** menu and change the ALRT's resource id to 129. Next, close the Info window and change the **DITL ID:** field in the ALRT window to 129. We have to create a DITL for the items in the alert and, since we've already got a DITL 128, we'll use 129 for the alert.

In general, it's a good idea to keep the ALRT and DLOG resource ids in sync with their respective DITL resource ids. Our DLOG 128 goes with DITL 128 and ALRT 129 goes with DITL 129.

Double-click the ALRT window inside the mini-desktop. A new DITL resource will appear. You'll add two items to the DITL. First, create a Button according to the specs in **Figure 18**.

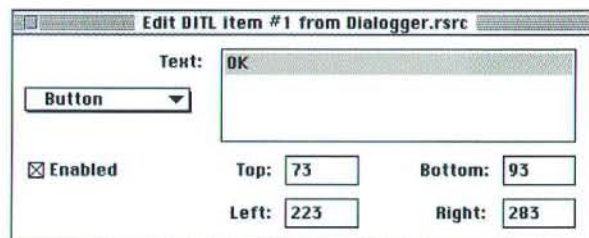


Figure 18. Specifications for the alert's OK button.

The best tool for text.

BEdit. The premier text &
HTML Editor for the Mac OS®

From those who know ...

"I've been using BEdit for about a month and now find it invaluable. It seems to offer just the tools I need not only for HTML coding (which is what I bought it for), but also for writing and research, which came as a surprise. As a devotee of light and flexible vs. bloated and unwieldy, I've taken to using BEdit for the daily grind of my writing chores. When I get bored, I just turn to 'grep' and toy with some complex search and replace expression! Thanks for your solid work."

*David Eddy Spicer
Assistant Director,
Case Program
John F. Kennedy
School of Government
Harvard University*

"I use BEdit every day and I'm continually blown away by the intelligent interface, the power user features like 'grep', search and replace across multiple files, the syntax coloring, the FTP open & save, the configurability, Internet Config, everything. Every time I think of something I'd like to see added to the program, I discover that it's already there. You guys really thought of everything! When I use other programs, I find myself wishing for the features that BEdit has."

*Chris Varosy
US Web/Dream Media*



<http://web.barebones.com/>



Bare Bones Software, Inc.

P.O. Box 1048, Bedford, MA 01730 • main 781-687-0700 • fax 781-687-0711

Next, create a Static Text item according to the specs in **Figure 19**. Be sure the **Enabled** check box is unchecked.

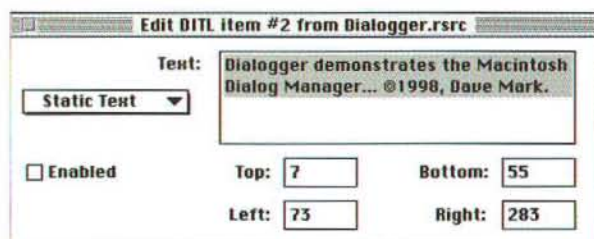


Figure 19. Specifications for DITL item #2.

Whew! That's it. Quit ResEdit, being sure to save your changes. Let's get to the project.

CREATING THE DIALOGGER PROJECT

Launch CodeWarrior and create a new project based on the MacOS:C/C++:Basic Toolbox 68k stationary. Turn off the **Create Folder** check box. Name the project Dialogger.mcp and place it in your Dialogger folder. Remove SillyBalls.c and SillyBalls.rsrc from the project; we will not be using these files. From the Finder, drag and drop your Dialogger.rsrc file into the project window. You also can remove the ANSI Libraries group from the project, because we won't need them, either.

Select **New** from the **File** menu to create a new window. Save it under the name Dialogger.c. Select **Add Window** from the **Project** menu to add Dialogger.c to the project. Your project window should look something like **Figure 20**.



Figure 20. Dialogger project window.

Rather than print the code here twice, we'll go straight to the walk-through. You can type in the code as we discuss it below and you will end up with the complete program, or you can save your fingers some effort and get the complete project from MacTech's ftp site <<http://ftp.mactech.com/src/>>.

WALKING THROUGH THE SOURCE CODE

Much of the Dialogger source code will look familiar to you from earlier programs. I'll keep the chatter to a minimum. You've seen some of these constants before. The rest I'll explain as we get to them. One convention you should be aware of is the 'i' convention. Just as we did with menu items, constants that reflect a dialog item start with the letter 'i'.

```
#include <Controls.h>
#include <Dialogs.h>
#include <Menus.h>
#include <Quickdraw.h>
#include <Sound.h>

#define kBaseResID 128
#define kAboutALRTid 129
#define kDialogResID 128

#define kVisible true
#define kMoveToFront (WindowPtr)-1L
#define kNoGoAway false
#define kSleep 7L

#define kFirstRadio 3

#define kOn 1
#define kOff 0
```

Each time you add an item to a dialog item list, the item is given a unique number. The OK button always gets number 1 and the Cancel button always gets number 2. ResEdit's DITL menu (which appears when you edit a DITL) allows you to renumber the items in a DITL, as well as turn the display of item numbers on and off. In general, it's a good idea to write down (or print out) each of your DITL item id's once you finish your ResEdit session and you start writing your code. I usually create the item #defines for my MENU and DITL resources right away so I won't forget them.

```
#define iAfghan 3
#define iElephant 4
#define iSquirrel 5

#define iShowPreview 7
#define iUserItem 8

#define kLeftMargin 5
#define kTopMargin 40

#define mApple kBaseResID
#define iAbout 1

#define mFile kBaseResID+1
#define iSettings 1
#define iQuit 3
```

Dialogger makes use of three globals. gDone is set to true when the program is ready to exit. gShowPreview corresponds to the Show preview checkbox in the Settings dialog. It is set to true whenever the check box is checked. We could avoid the use of a global by using the same initial setting for Show preview each time we enter the routine that handles the dialog box. By using a global, however, the setting of the check box survives, even after the dialog is dismissed.

```
/****** Globals *****/
Boolean gDone, gShowPreview = true;
```


Power Too Abundant to Meter!

neo•logic®

Powering Development of Object-Oriented Applications



Powerful

No Runtime Fees

Cross Platform

Proven

Save \$100

NeoAccess is the most powerful object-oriented database engine available. It displays electrifying performance—up to ten times that of its competitors. Behind its elegant programming interface is a fully optimized relational query engine tuned for short access times and an object cache with automatic garbage collection for nearly instantaneous access to previously used objects. All in a memory footprint as small as 150K.

Get the power of NeoAccess, and avoid the expense and administrative hassle of feeding the runtime fees meter. You pay one affordable price no matter how many copies of your application you sell or use.

Others may promise cross-platform development tools—NeoLogic delivers. NeoAccess is a set of C++ classes designed for use with popular compilers and application frameworks on Windows®, Macintosh®, and Unix™ platforms. Source code is included so it can even be used with custom frameworks.

Thousands of developers, including America Online® and Claris®, have already found that NeoAccess enabled them to build fast, powerful internet applications in record time. That's why there are more copies of NeoAccess based applications on end-user machines than any other object database backend. Tap the power!

Order Now!

M-F 9AM to 5PM Pacific: **1-800-919-6353**
For Information and Customer Service: **1-510-524-5897**

neoAccess®

Cross-Platform Object Database Engine

NeoLogic Systems, Inc.

1450 Fourth St., Suite 12, Berkeley, CA 94710
v. 510.524.5897 f. 510.524.4501 neologic@neologic.com

Order Directly from Our Web Site!
Download the Architectural Overview

www.neologic.com

The same thinking holds true for gCurrentPICT. This global tells us which picture we're currently looking at, ensuring that the dialog's Show preview brings up the current picture of my pet Fred.

```
short gCurrentPICT = kBaseResID;
```

As always, here's a complete list of function prototypes.

```
/****** Functions *****/  
void ToolBoxInit( void );  
PicHandle LoadPICT( short picID );  
void CreateWindow( void );  
void MenuBarInit( void );  
void EventLoop( void );  
void DoEvent( EventRecord *eventPtr );  
void HandleMouseDown( EventRecord *eventPtr );  
void HandleMenuChoice( long menuChoice );  
void HandleAppleChoice( short item );  
void HandleFileChoice( short item );  
void DoUpdate( EventRecord *eventPtr );  
void DoDialog( void );  
void FlipControl( ControlHandle control );  
void DrawPreview( DialogPtr dialog, short picID );  
void SwitchPICT( void );
```

The following two routines are part of System 7, and won't work with earlier versions of the operating system.

main() initializes the Toolbox, then sets up the menu bar and creates the My Pet Fred window. Once that's done, the main event loop is entered.

```
/****** main *****/  
  
void main( void )  
{  
    ToolBoxInit();  
    MenuBarInit();  
  
    CreateWindow();  
  
    EventLoop();  
}
```

Nothing new here...

```
/****** ToolboxInit */  
void ToolBoxInit( void )  
{  
    InitGraf( &qd.thePort );  
    InitFonts();  
    InitWindows();  
    InitMenus();  
    TEInit();  
    InitDialogs( NULL );  
    InitCursor();  
}
```

LoadPICT() uses GetPicture() to load a PICT resource. If the PICT can't be found, beep once, then exit.

```
/****** LoadPICT *****/  
PicHandle LoadPICT( short picID )  
{  
    PicHandle pic;  
  
    pic = GetPicture( picID );
```



```

if ( pic == NULL )
{
    SysBeep( 10 ); /* Couldn't load the PICT resource!!! */
    ExitToShell();
}
return pic;
}

```

CreateWindow() loads the current PICT, then uses the Rect that defines its border to define the size of a new window. The idea here is to create a window exactly big enough to hold the entire PICT. Of course, this wouldn't be a good idea if the PICT was bigger than the entire screen. Hmmm... Is this an idea for a column on scroll bars? We'll see...

```

/***** CreateWindow *****/
void CreateWindow( void )
{
    WindowPtr    window;
    PicHandle    pic;
    Rect         r;

    pic = LoadPICT( gCurrentPICT );

    r = (**pic).picFrame;

```

We really don't care where on the screen the PICT is, we just care about its size. We'll use OffsetRect() to normalize the Rect, moving it so it's the size of the PICT, but so its upper-left corner is at the point (kLeftMargin, kTopMargin).

```

OffsetRect(&r, kLeftMargin - r.left, kTopMargin - r.top);

```

Next, we use NewWindow() to create a new window. We could have used GetNewWindow() but we'd have to change the window's size. Whatever floats your boat...

```

window = NewWindow( NULL, &r, "\pMy Pet Fred", kVisible,
    noGrowDocProc, kMoveToFront, kNoGoAway, 0L );

```

If the window couldn't be created, beep once, then exit.

```

if ( window == NULL )
{
    SysBeep( 10 ); /* Couldn't load the WIND resource!!! */
    ExitToShell();
}

```

Since we created the window as kVisible, this call to ShowWindow() is redundant. I prefer to leave it in, so it's very obvious when I revisit the code at 3 in the morning.

```

ShowWindow( window );
SetPort( window );
}

```

MenuBarInit() does the usual, loading the MBar resource, adding the apple items to the menu, then drawing the menu bar.

```

/***** MenuBarInit *****/
void MenuBarInit( void )
{
    Handle    menuBar;
    MenuHandle menu;

    menuBar = GetNewMBar( kBaseResID );
    SetMenuBar( menuBar );

```

```

menu = GetMenuHandle( mApple );
AppendResMenu( menu, 'DRVR' );
DrawMenuBar();
}

```

EventLoop() and DoEvent() should be familiar to you. No big changes.

```

/***** EventLoop *****/
void EventLoop( void )
{
    EventRecord event;
    gDone = false;
    while ( gDone == false )
    {
        if ( WaitNextEvent( everyEvent, &event, kSleep, NULL ) )
            DoEvent( &event );
    }
}

/***** DoEvent */
void DoEvent( EventRecord *eventPtr )
{
    char theChar;
    switch ( eventPtr->what )
    {
        case mouseDown:
            HandleMouseDown( eventPtr );
            break;
        case keyDown:
        case autoKey:
            theChar = eventPtr->message & charCodeMask;
            if ( (eventPtr->modifiers & cmdKey) != 0 )
                HandleMenuChoice( MenuKey( theChar ) );
            break;
        case updateEvt:
            DoUpdate( eventPtr );
            break;
    }
}

```

As you've seen in other programs, HandleMouseDown() processes the latest mouseDown event.

```

/***** HandleMouseDown */
void HandleMouseDown( EventRecord *eventPtr )
{
    WindowPtr window;
    short thePart;
    long menuChoice;

```

FindWindow() tells you which window, and which part of the window, the mouseDown occurred in.

```

thePart = FindWindow( eventPtr->where, &window );

```

Depending on the part of the window the mouseDown occurred in, the event is handled accordingly.

```

switch ( thePart )
{
    case inMenuBar:
        menuChoice = MenuSelect( eventPtr->where );
        HandleMenuChoice( menuChoice );
        break;
    case inSysWindow:
        SystemClick( eventPtr, window );
        break;
    case inContent:
        SelectWindow( window );
        break;
    case inDrag:
        DragWindow( window, eventPtr->where,
            &qd.screenBits.bounds );
        break;
}
}

```


"Without a doubt, the Premiere Resource Editor for the Mac OS ... A wealth of time-saving tools."

— MacUser Magazine Eddy Awards

"A distinct improvement over Apple's ResEdit."

— MacTech Magazine

"Every Mac OS developer should own a copy of Resorcerer."

— Leonard Rosenthol, Aladdin Systems

"Without Resorcerer, our localization efforts would look like a Tower of Babel. Don't do product without it!"

— Greg Galanos, CEO and President, Metrowerks

"Resorcerer's data template system is amazing."

— Bill Goodman, author of *Smaller Installer* and *Compact Pro*

"Resorcerer Rocks! Buy it, you will NOT regret it."

— Joe Zobkiw, author of *A Fragment of Your Imagination*

"Resorcerer will pay for itself many times over in saved time and effort."

— MacUser review

"The template that disassembles PICT's is awesome!"

— Bill Steinberg, author of *Pyro!* and *PBTools*

"Resorcerer proved indispensable in its own creation!"

— Doug McKenna, author of *Resorcerer*



Resorcerer® 2

Version 2.0

The Resource Editor for the Mac™ OS Wizard

ORDERING INFO

Requires System 7.0 or greater,
1.5MB RAM, CD-ROM

Standard price: \$256 (decimal)

Website price: \$128 - \$256

(Educational, quantity, or
other discounts available)

Includes: Electronic documentation
60-day Money-Back Guarantee
Domestic standard shipping

Payment: Check, PO's, or Visa/MC
Taxes: Colorado customers only

Extras (call, fax, or email us):
COD, FedEx, UPS Blue/Red,
International Shipping

MATHEMAESTHETICS, INC.
PO Box 298
Boulder, CO 80306-0298 USA
Phone: (303) 440-0707
Fax: (303) 440-0504
resorcerer@mathemaesthetics.com

New
in
2.0:

- Very fast, HFS browser for viewing file tree of all volumes
- Extensibility for new Resorcerer Apprentices (CFM plug-ins)
- New AppleScript Dictionary ('aete') Apprentice Editor
- MacOS 8 Appearance Manager-savvy Control Editor
- PowerPlant text traits and menu command support
- Complete AIFF sound file disassembly template
- Big-, little-, and even mixed-endian template parsing
- Auto-backup during file saves; folder attribute editing
- Ships with PowerPC native, fat, and 68K versions

- Fully supported; it's easier, faster, and more productive than ResEdit
- Safer memory-based, not disk-file-based, design and operation
- All file information and common commands in one easy-to-use window
- Compares resource files, and even **edits your data forks** as well
- Visible, accumulating, editable scrap
- Searches and opens/marks/selects resources by text content
- Makes global resource ID or type changes easily and safely
- Builds resource files from simple Rez-like scripts
- Most editors DeRez directly to the clipboard
- All graphic editors support screen-copying or partial screen-copying
- Hot-linking Value Converter for editing 32 bits in a dozen formats
- Its own 32-bit List Mgr can open and edit very large data structures
- Templates can pre- and post-process any arbitrary data structure
- Includes nearly 200 templates for common system resources
- TMPLs for Installer, MacApp, QT, Balloons, AppleEvent, GX, etc.
- Full integrated support for editing color dialogs and menus
- Try out balloons, 'ictb's, lists and popups, even create C source code
- Integrated single-window Hex/Code Editor, with patching, searching
- Editors for cursors, versions, pictures, bundles, and lots more
- Relied on by thousands of Macintosh developers around the world

To order by credit card, or to get the latest news, bug fixes, updates, and apprentices, visit our website...

www.mathemaesthetics.com

HandleMenuChoice() handles menu selections from either the Apple or File menus.

```

/***** HandleMenuChoice *****/
void HandleMenuChoice( long menuChoice )
{
    short menu;
    short item;

    if ( menuChoice != 0 )
    {
        menu = HiWord( menuChoice );
        item = LoWord( menuChoice );

        switch ( menu )
        {
            case mApple:
                HandleAppleChoice( item );
                break;
            case mFile:
                HandleFileChoice( item );
                break;
        }
        HiliteMenu( 0 );
    }
}

```

HandleAppleChoice() is the same as in previous programs.

```

/***** HandleAppleChoice *****/
void HandleAppleChoice( short item )
{
    MenuHandle appleMenu;
    Str255      accName;
    short      accNumber;

    switch ( item )
    {
        case iAbout:
            NoteAlert( kAboutALRTid, NULL );
            break;
        default:
            appleMenu = GetMenuHandle( mApple );
            GetMenuItemText( appleMenu, item, accName );
            accNumber = OpenDeskAcc( accName );
            break;
    }
}

```

HandleFileChoice() starts off by declaring a totally useless (and unused) variable. Feel free to delete this line of code. I don't know what I was thinking!

```

/***** HandleFileChoice *****/
void HandleFileChoice( short item )
{
    short newPictId;

```

If Settings was selected, call DoDialog() to put up the dialog box. If Quit was selected, set gDone to true which will cause the program to drop out of the main event loop.

```

switch ( item )
{
    case iSettings:
        DoDialog();
        break;
    case iQuit:
        gDone = true;
        break;
}
}

```

DoUpdate() gets called whenever the Window Manager generates an updateEvt, asking Dialogger to redraw the contents of the My Pet Fred window.

```

/***** DoUpdate */
void DoUpdate( EventRecord *eventPtr )
{
    PicHandle pic;
    WindowPtr window;
    Rect r;

```

The WindowPtr is embedded in the event's message field. We'll need this in a bit.

```

window = (WindowPtr)eventPtr->message;

```

Since the window was created the exact size of the current picture, we'll load the PICT, make window the current port, then draw the PICT in the window.

```

pic = LoadPICT( gCurrentPICT );

SetPort( window );

BeginUpdate( window );

r = window->portRect;
DrawPicture( pic, &r );

EndUpdate( window );
}

```

DoDialog() is where the real action is.

```

/***** DoDialog */
void DoDialog( void )
{
    DialogPtr dialog;
    Boolean dialogDone = false;
    short itemHit, itemType;
    Handle itemHandle;
    Rect itemRect;
    short curRadioButton;

```

GetNewDialog() loads a DLOG resource (as well as its associated DITL resource) with the id specified in the first parameter. The second parameter allows you to designate a filter function that will get called repeatedly as events occur inside your dialog. (We won't use this filter function here. Maybe in a future column). The third parameter determines whether the dialog's window appears in front of all other windows, just like the corresponding parameter in GetNewWindow().

```

dialog = GetNewDialog( kDialogResID, NULL, kMoveToFront );

```

Just as you would with a window, make the dialog visible and make it the current port. As you'll see, DialogPtrs and WindowPtrs are quite similar and DialogPtrs can be passed to the same routines you'd normally pass a WindowPtr to.

```

ShowWindow( dialog );
SetPort( dialog );

```

SetDialogDefaultItem() tells the Dialog Manager which button is the default button (normally the OK button). The default button is the button you'd like activated when the user hits the return key or presses enter. The Dialog Manager will make sure to draw a bold outline around the default button, making it easy for the user to see.

```

SetDialogDefaultItem( dialog, ok );

```


SetDialogCancelItem() performs a similar function, allowing you to specify the button that gets activated when the user types (normally the Cancel button). ok and cancel are predefined in a #include file to be 1 and 2 respectively.

```
SetDialogCancelItem( dialog, cancel );
```

This next calculation tells us which of our three radio buttons should be turned on. Radio buttons always travel in sets. Just like the channel selector buttons on your car radio, one button is always selected, and the others always off. When a radio button is selected its circle is filled in, like the Elephant radio button in **Figure 21**.



Figure 21. A set of radio buttons.

```
curRadioButton = gCurrentPICT - kBaseResID + kFirstRadio;
```

GetDialogItem() uses a dialog item's id to retrieve its type, a handle to it, and its bounding Rect. We'll cast the retrieved handle to a ControlHandle, then use the Control Manager routine SetControlValue() to set the value of the radio button control to kOn. As you can see, radio buttons have two legal values, on (1) and off (0). All controls have a limited range of values. Buttons, radio buttons, check boxes, and scroll bars are all examples of controls.

```
GetDialogItem( dialog, curRadioButton, &itemType,
               &itemHandle, &itemRect );
SetControlValue( (ControlHandle)itemHandle, kOn );
```

If the Show preview check box is supposed to be checked, we'll use GetDialogItem() and SetControlValue() to turn it on. By default, all dialog item controls are off.

```
if ( gShowPreview )
{
    GetDialogItem( dialog, iShowPreview, &itemType,
                  &itemHandle, &itemRect );
    SetControlValue( (ControlHandle)itemHandle, kOn );
}
```

Our last step in preparing our dialog box is to call our own DrawPreview() function. DrawPreview() checks to see if the Show preview check box is checked. If so, it will draw the miniature picture of My Pet Fred. Otherwise, DrawPreview() will erase any existing preview, then exit.

```
DrawPreview( dialog, curRadioButton +
             kBaseResID - kFirstRadio );
```

IS YOUR SOFTWARE
READY TO "RHAP"?

LICENSER KIT®

INTRODUCTORY PRICE

\$1995

- Create floating network licenses
- Tie license to host name
- Tie license to user name
- Tie to host ID
- Allow as many users as desired
- Have unlimited number of licenses on a network
- Add licenses on the fly
- Automatically unlicense if application is moved or copied
- Generate expiring licenses for "full strength" demos
- Works on **Yellow Box for Windows** as well as MACH/Rhapsody
- Comes with sample app and license key generator

FROM YOUR FRIENDS AT

Caffeine
SOFTWARE



www.stone.com/Licenser/ info@stone.com (505)-345-4800

OK. Now our setup work is done. We've set our controls to their initial values, and performed any necessary drawing. This is typical of dialog management. Now we'll enter the dialog loop, which will bring our dialog to life. The dialog loop keeps going until we set dialogDone to true. We'll do this when either the Cancel or OK buttons is pressed.

```
while ( ! dialogDone )
{
```

King of all Dialog Manager routines, ModalDialog() grabs an event from the event queue, calls the filter function (if you've specified one), then processes the event. ModalDialog() sets itemHit to the item clicked (if the event was a mouseDown).

```
ModalDialog( NULL, &itemHit );
```

Now all you have to do is figure out what to do with itemHit.

```
switch( itemHit )
{
```

If the click was in the OK or Cancel buttons, set dialogDone to true.

```
case ok:
case cancel:
    dialogDone = true;
    break;
```


If the click was in the Show preview check box, we'll flip the value of the check box (turn it on if off, off if it was on) then call `DrawPreview()` to redraw the preview area accordingly.

```
case iShowPreview:
    GetDialogItem( dialog, iShowPreview, &itemType,
        &itemHandle, &itemRect );
    FlipControl( (ControlHandle)itemHandle );

    DrawPreview( dialog, curRadioButton +
        kBaseResID - kFirstRadio );
    break;
```

If one of the radio buttons was clicked, we first check to make sure it wasn't the lit one. If the lit one was clicked, there's no reason to do anything.

```
case iAfghan:
case iElephant:
case iSquirrel:
    if ( curRadioButton != itemHit )
    {
```

If a new one was clicked, turn off the current button, then turn on the new one. Always turn off the old button before you turn on the new one. That way you won't ever have two radio buttons lit at the same time. This can be jarring to the user.

```
        GetDialogItem( dialog, curRadioButton, &itemType,
            &itemHandle, &itemRect );
        FlipControl( (ControlHandle)itemHandle );

        GetDialogItem( dialog, itemHit, &itemType,
            &itemHandle, &itemRect );
        FlipControl( (ControlHandle)itemHandle );
```

Next, make `curRadioButton` reflect the new current radio button, then redraw the preview.

```
        curRadioButton = itemHit;

        DrawPreview( dialog, curRadioButton +
            kBaseResID - kFirstRadio );
    }
    break;
}
```

Once we drop out of the main dialog loop, we'll make the dialog window invisible. Even though it's invisible, we'll still have access to its controls, so we can change some things then make it visible again if we want.

```
HideWindow( dialog );
```

If the dialog was dismissed with a click in Cancel, we don't care about any changes made since the dialog was opened. If the click was in OK (if the user pressed return or enter the Dialog Manager is nice enough to simulate the appropriate click for us), we'll save the current value of the Show preview check box in `gShowPreview`.

```
if ( itemHit == ok )
{
    GetDialogItem( dialog, iShowPreview, &itemType,
        &itemHandle, &itemRect );
    if ( GetControlValue( (ControlHandle)itemHandle )
        == kOn )
        gShowPreview = true;
    else
        gShowPreview = false;
}
```

Next, if the state of the radio buttons has changed, we'll call the routine `SwitchPICT()` to switch the My Pet Fred window to reflect our new choice for a domestic destructive device.

```
if ( gCurrentPICT != curRadioButton +
    kBaseResID - kFirstRadio )
{
    gCurrentPICT = curRadioButton +
        kBaseResID - kFirstRadio;
    SwitchPICT();
}
```

Either way, once we're done, we call `DisposeDialog()` to free up all memory allocated for the dialog. If we hadn't made the dialog invisible first, this would have done that for us.

```
DisposeDialog( dialog );
}
```

`FlipControl()` calls `GetControlValue()` to retrieve the current value of the control, then uses `!` and `SetControlValue()` to flip its value.

```
/* FlipControl */
void FlipControl( ControlHandle control )
{
    SetControlValue( control, ! GetControlValue( control ) );
}
```

`DrawPreview()` first checks to see if the Show preview check box is off.

```
/* DrawPreview */
void DrawPreview( DialogPtr dialog, short picID )
{
    PicHandle pic;
    short itemType;
    Handle itemHandle;
    Rect itemRect;

    GetDialogItem( dialog, iShowPreview, &itemType,
        &itemHandle, &itemRect );
    if ( GetControlValue( (ControlHandle)itemHandle ) == kOff )
    {
```

If so, the preview area is erased and `DrawPreview()` returns.

```
        GetDialogItem( dialog, iUserItem, &itemType,
            &itemHandle, &itemRect );
        EraseRect( &itemRect );
        return;
    }
```

If the check box is checked, the current picture is drawn in the rectangle specified by the user item. As you can see, the user item comes in handy for marking a rectangle in your dialog box. If you want to move the preview area, just go into ResEdit and drag it around or resize it and your changes are reflected next time you run your program.

```
    pic = LoadPICT( picID );

    GetDialogItem( dialog, iUserItem, &itemType,
        &itemHandle, &itemRect );
    FrameRect( &itemRect );

    InsetRect( &itemRect, 1, 1 );
    DrawPicture( pic, &itemRect );
}
```


SwitchPICT() deletes the front window, then creates a new window. Since CreateWindow() uses the current picture to determine the size and contents of its window, the newly selected pet Fred candidate will appear.

```

/***** SwitchPICT */
void SwitchPICT( void )
{
    WindowPtr window;

    window = FrontWindow();
    DisposeWindow( window );

    CreateWindow();
}

```

RUNNING DIALOGGER

Save your source code and you're ready to roll. When you run the project, a window should appear, just big enough to hold PICT 128. If you made your PICTs too small, go back and fix them. **Figure 22** shows my window.



Figure 22. My Pet Fred.

Go to the File menu and select Settings (**Figure 23**). Play with all the dialog items. Change some things, then select Cancel, verifying that things stay the same the next time you go into the dialog box. Uncheck the Show preview check box to see what happens. Play, play, play!

Want to get up on a soap box?
Send your opinions to
<mailto:letters@mactech.com>

New Version 7.0!

Snowbound Software

HIGH PERFORMANCE

Imaging, Annotation & Plug-in Development Tools

RasterMaster 7.0 Powerful, multi-platform raster imaging support for **60+ Raster Formats**. **Familiar API and well-proven technology** in use worldwide by companies such as BP, Chase Manhattan, Ford, Gannett, HP, IMSI, Kodak, LEXIS-NEXIS, Polaroid, Toyo, Unisys, Xerox and more.

Formats: TIF, JPEG, MO:DCA, CALS, PNG, DICOM, Flashpix, BMP, BRK, G3, G4, GIF, JED, KFX, MAC, IOCA, PCX, TGA, WFX and more. Features include: (* New for Version 7)

• Compression	• CMYK 4 Plane Support *
• Image Processing	• 10-16 bit Gray Scale *
• Despeckle & Deskew	• Anti-Aliasing in Silver *
• Multipage TWAIN Scanning	• Fit to Width & Height *
• Pan, Zoom, Scroll	• Tiled Image Support *
• Image Editing	• Medical Imaging
• Auto Aspect Ratio	• Animated GIF *
• Alpha Channel Support *	• Image Encryption * & more

RasterNote 2.0 Annotation/Redlining toolkit. Features include: Sticky Notes, Lines, Ellipses, Freehand Drawing, Polygons, Highlighting, Redaction, Multiple Annotation layers.

RasterNet 2.0 Plug-in for Netscape and Internet Explorer. Read 60+ Raster Formats! Simple drop in library makes it easy to enable standard browsers to read your Internet/Intranet pages.

Platforms Include: Win 95/NT, Win 3.x, Mac 68K/PPC, UNIX, Alpha, OS/2
Languages: DLL, VBX, ActiveX/OCX, Delphi, FoxPro, PB, VC++

CALL 617-630-9495 OR VISIT US AT
www.snowbnd.com Email: Sales@m.snowbnd.com
Snowbound Software, 29 Crafts St, Suite 380, Newton, MA 02158 Fax: 617-630-0210



Figure 23. The Settings dialog box.

TILL NEXT MONTH

Wow, these columns are getting long! That's the price we pay as we dive deeper into the Toolbox, I guess. Well, I'll see you next month. Till then, read the Dialog and Control Manager chapters in *Inside Macintosh: Macintosh Toolbox Essentials*. **MT**

by Michael Marchetti

NeoAccess 5.0 Revealed

Taking a look at this high-performance object database management system

INTRODUCTION

As a developer of custom software, I have the opportunity to work on a variety of projects, from the design phase all the way through the maintenance cycle. In 1989, ITA designed and implemented a program which stored and retrieved a number of different kinds of data. Its data storage mechanism consisted of MacApp's dynamic arrays, stored in a simple custom file format. Over the last eight years our requirements changed in a number of ways. The sheer volume of data increased greatly, resulting in increased memory usage, longer search times, and longer program startup times. New features required us to implement new access mechanisms, adding complexity to our original design. Finally, we were asked to port the program to Windows. Our Mac-centric data storage used Pascal strings and assumed big-endian integer and floating-point representations. We were looking at a major overhaul. After evaluating a number of storage mechanisms, we chose NeoAccess.

WHY NEOACCESS?

At first, the idea of using a full-featured object database management system seemed like overkill. As we evaluated NeoAccess, however, its advantages became apparent.

NeoAccess does not load all of a database's data at once. Instead, objects are brought into memory on demand and cached in a bounded amount of memory. A reference-counting mechanism allows NeoAccess to effectively manage the cache with minimal developer involvement. This provides developers with a number of benefits.

- **Memory requirements:** NeoAccess can store large amounts of data using a bounded amount of memory as an object cache. Objects are reference-counted, allowing NeoAccess to effectively manage the cache with minimal developer involvement.
- **Startup time:** Documents open faster because not all the data is loaded at once.
- **Flexibility:** NeoAccess allows the application developer to easily change the database format and still support reading, converting, and even creating databases using older formats.
- **Access mechanisms:** NeoAccess supports relational queries as well as referential (object network) access. Database developers can choose the most appropriate method for each situation.
- **Cross-platform compatibility:** NeoAccess is available for Macintosh, Windows, and Unix, with a uniform feature set across all supported platforms. Database files are written in a canonical form that can be read on any platform.
- **Object-oriented design:** NeoAccess is written entirely in object-oriented C++. Databases store and retrieve true C++ objects.

Michael Marchetti <mmarchetti@itainc.com> develops Macintosh software at ITA, Inc., a provider of custom software solutions located in Rochester, New York.

- **Framework support:** NeoAccess application, document, stream, and persistent object classes are integrated into the frameworks available for each platform. NeoAccess also includes a "standalone mode" for use with custom frameworks or without any framework.
- **Multi-threading:** NeoAccess can be safely used in a multi-threaded environment and takes advantage of asynchronous I/O operations to improve throughput.
- **Capacity:** NeoAccess can store up to 4 billion objects per database. The Mac OS limits database files to 4GB; NeoAccess can be configured to use 63-bit file offsets on platforms that support larger files.

FIRST IMPRESSIONS

The NeoAccess 5.0 package I reviewed is for Macintosh only (the multi-platform version has identical core code but includes source and project files specific to the other platforms). The package consists of one CD and two manuals. The installation procedure involves simply copying the contents of the CD to a hard drive. The CD includes:

- Full source code to the database engine.
- The complete manual in PDF form.
- Several add-on features (called "Extras"), with source code and PDF documentation.
- Two demo programs with source code and CodeWarrior projects.

The two demo programs come pre-built in a number of configurations, with SYM files. This means you can use the debugger to learn how a NeoAccess program and the NeoAccess engine work. The demos come in different flavors for different frameworks. Version 5.0 supports *standalone* (no framework) and *PowerPlant*. This is a change from version 4.0, which included support for MacApp 3.3 and TCL.

The demo programs are NeoBench and Laughs. NeoBench is a benchmarking program to test the speed of the database engine. Laughs illustrates most of the core constructs in NeoAccess, including inheritance, part managers, blobs, strings, and indices. The documentation includes a tutorial which describes the classes used in the Laughs application and how they interact.

TECHNICAL INTRODUCTION

The main storage container in NeoAccess is the database, modeled by the class `CNeoDatabase`. A database is a file containing a set of objects. These objects are partitioned into classes which correspond to the subclasses of `CNeoPersist` defined in the program. (Application-specific classes inherit their persistence properties from `CNeoPersist`.) The program supplies NeoAccess with information about the inheritance relationships between classes. This enables the program to limit a search to objects of one specific class or allow it to range over a class and all subclasses.

TestTrack

Your Total Bug Tracking Solution



**"If you're a developer, you've got to get TestTrack.
Your bugs will hate you for it!"** Cool Tool of the Day

Discover the tool today's top software developers are using to improve the quality of their **Macintosh and Windows** applications — **TestTrack**.

- Track bugs, feature requests, problems, customer information, and more.
- New! E-mail notifications — SMTP and MAPI.
- New! Import bugs via e-mail automatically.
- Produce concise reports.
- Multiple users, full security — link engineers, testers, managers, even tech writers.
- New! Improved help desk support.
- New! Deferred defect numbering, external attachment storage, customer bug histories, and much, much, more.
- Automatically route bugs to team members.
- Track the history of each bug.
- Save time and improve tech support by giving Solo Bug, TestTrack's stand-alone bug reporter, to your customers.

Only \$169! 2+ for \$149 each!
To order call 888-683-6456
or 513-683-6456



Download our demo today!
<http://www.seapine.com>

sales@seapine.com
<http://www.seapine.com>

**Seapine
Software**

Each class has a defined set of attributes (corresponding to its persistent data members) which are present in each object of that class in the database. Access to these attributes is through the virtual functions `setValue()` and `getValue()`, which the NeoAccess manual correctly describes as "the mother of all accessor functions." They provide access to every persistent attribute of an object, and perform type coercion as needed (if, for example, the requested attribute is stored as a Pascal string and requested as a C string).

Indices

Each class has a set of indices associated with it. An index is a list of every object of that class, sorted in order of a particular attribute. Indices make it possible to find objects efficiently and to iterate through a set of objects in order.

Every class has at least one index, known as the **primary index**. The default primary index sorts objects by object ID (a unique 32-bit integer assigned when the object is added to the database). All other indices are known as **secondary indices**. These are mappings from attribute values to object IDs. When searching a secondary index, NeoAccess uses the primary index to actually locate objects once their IDs are known.

When the program attempts to locate an object with a particular attribute value, the NeoAccess query optimizer looks for an index on that attribute. If an index is found, a binary

search algorithm is used to locate the object. If the index does not exist, then a linear search algorithm must be used. When the program requests an iterator in order of a particular attribute, NeoAccess looks for an index on that attribute, then iterates through the index.

Secondary indices can be added and removed at runtime. This is useful if access patterns tend to change over time or are based on user preferences. Indices that are not frequently used can be removed to save space, while new indices can be created to index those attributes that are most frequently searched.

Part Managers

NeoAccess also provides a construct known as a part manager. Part managers can be thought of as persistent lists, implementing a one-way, one-to-many relationship. It is also useful to think of part managers as secondary indices that index only a subset of the objects in a class (namely those objects that have been explicitly added to the list).

For example, in a file system database, a directory object might contain a part manager which lists the directory contents in order of name. In fact, it might have several part managers sorting the contents by name, file size, and modification date. Keeping all of those part managers synchronized in the presence of system crashes and other anomalies could be problematic. One solution to the problem is to build the part managers dynamically when the directory is accessed. This can be easily done using queries.

Queries

A query is a persistent object which contains multiple part managers sorted in different orders. Objects are placed in the part managers by executing a query. In our example, we could assign each item a parent directory ID, and make a query to select those items with the desired parent ID. Since queries are persistent, both the selection criterion and the list of objects can be saved in the database. It is also simple to execute a query, use the results, and then discard the resulting lists.

Selection

NeoAccess defines a specialized mechanism for expressing selection criteria. There are a number of "selection key" classes which inherit from a base class of **CNeoSelect**. Most of them fall into the category of type-specific keys. **CNeoLongSelect** is used for long integer attributes, **CNeoStringSelect** for string attributes, and so on. These keys can be configured to search for an exact match or use another criterion such as "less than x" or "greater than x". There are also "complex keys," which combine multiple criteria into a single key. These include Boolean AND and OR as well as value ranges. Selection keys are used in all of the calls which retrieve data:

- **findObject** locates an object in the database matching the key.
- **getIterator** returns an iterator object, which the caller can use to walk through a set of objects in the database or in a part manager.

- **CNeoQuery** objects construct one or more sorted lists of objects matching the key.

Blobs

Some entities, such as image and movie data, cannot be easily represented with objects. NeoAccess provides a type of object, known as a blob, that simply stores a chunk of data. This is useful for image data, movies, and long text strings. Blobs can be indexed like any other field by using the **CNeoBlobIndex** class included with NeoAccess.

Dynamic Objects

Another optional component of NeoAccess is the DynaObject facility. When it is enabled through a compile-time flag, applications can add and remove attributes from persistent object classes at runtime and even create new classes on the fly. NeoAccess automatically maintains a prototype object for each persistent class and creates new instances from the prototype.

What's New

NeoLogic is constantly working to improve NeoAccess. Version 5.1 was just released when this article was written. It includes several new features and performance improvements.

Iterators can now be configured to keep track of the total number of entries and current position in the collection. This makes it much easier to implement scrolling lists efficiently.

The **distributed object facility** allows location-independent access to objects. This makes it possible to treat multiple databases as if they were one. For example, it is now possible to create an index in one database (the "host") to index objects stored in other databases (the "targets"). NeoAccess automatically opens and closes the target databases as needed. Part managers, queries, and swizzlers also implement object references that can refer to objects in other databases.

A general-purpose **test harness** allows developers to test their database code by subclassing a generic test class. Tests can be grouped into suites and scheduled to run sequentially or in random order, once or multiple times. The test harness has a command-line parser that passes parameters to the tests, allowing different options to be selected at runtime.

PERFORMANCE

I ran the NeoBench program on a PowerMac 7200/120 with VM off and a 256K system disk cache. In a 1500K partition, NeoBench had about 1200K free and used half (600K) as an object cache. I ran two sets of tests: one with 2,000 objects and one with 10,000 objects. This yielded the following performance results (in operations per second):

	2,000 objects	10,000 objects
Insert	1,030	1,140
Locate Randomly	14,600	11,000
Locate Serially	148,000	146,000
Change	1,140	1,220
Delete	7,170	8,800

Notes

1. The insert, change, and delete phases include committing changes to the database file.
2. Enabling debugging code decreases engine performance by about one-third for most operations.
3. Version 5.1, just released, has performance optimizations beyond those used in this test.

As you can see, NeoAccess achieves very fast access times. Like many benchmarks, NeoBench does not necessarily reflect the performance of real applications. It does a good job of testing the speed of the primary index, iteration, and object I/O. However, the persistent object class in NeoBench does not have any secondary indices. Each additional index slows down the insert, change, and delete operations. In addition, locating objects in a secondary index is slower than using the primary index.

DOCUMENTATION

NeoAccess does have a rather steep learning curve; new developers are presented with around 60,000 lines of code totaling 2MB. The documentation copes with this volume of information in three ways. First, the manual contains an introductory section explaining the concepts we have seen here. Second, most of the manual consists of the class reference. Third, the manual includes a tutorial section that explains the internal workings of the Laughs demo program, one section at a time. Since Laughs uses a large subset of NeoAccess, this section is useful as a quick reference with code examples.

Users of previous versions will notice significant improvements in the documentation. In particular, the introductory section now includes instructions and code snippets showing how to use the basic features of NeoAccess. The reference section includes more background information on a number of classes.

As more complex constructs are introduced (part managers, blobs, dynamic objects, etc.) you will have to override more NeoAccess functions in your classes. Although the documentation does a pretty good job of explaining what to override, it remains a tedious and error-prone process. NeoLogic is planning to simplify this procedure in a future release.

POWERKEY[®] PRO

The Programmable Power Strip

Take control of your Mac system!

Start up your entire system with a keystroke, or when the phone rings, or every morning, or in the middle of the night each Thursday. Switch outlets, open documents or run AppleScripts. Let your computer do those boring backups while you sleep.

PowerKey Pro's scheduling software lets you easily create powerful events to control your system. It's fully customizable to work the way you do. Put this power to work for you today!

Model 200

- Two groups of switched outlets
- Telephone ring detection
- Add Server Restart Option to automatically restart crashed servers

Model 600

All the features of Model 200, plus...

- All six outlets are individually switched
- Front panel switches and indicator lights
- Answer the phone and run events with custom dialing tones
- Server Restart Option included free (for a limited time)

800-769-3773 www.sophisticated.com

SOPHISTICATED CIRCUITS^{INC.}

Copyright ©1997 Sophisticated Circuits, Inc. PowerKey is a registered trademark of Sophisticated Circuits, Inc. Macintosh and Mac are registered trademarks of Apple Computer, Inc.

To use some of the engine's capabilities, developers must modify the `NeoTypes.h` include file. It contains a number of compile-time flags controlling a huge amount of functionality, as well as common type declarations. It would be nice if this file was split into a file containing the type declarations and a "developer-modifiable" file containing the compile flags which we could keep under source control for different projects.

DEBUGGING

Debugging NeoAccess programs can be complicated. There are a lot of assertions built into the database engine to check all sorts of things: parameter values, usage, and database consistency. However, even with those checks, it is still easy to write code that fails without producing an error message. One particular problem is when objects don't sort correctly. Usually this means that an index is missing, or indexed attributes were modified without updating the index. It would be helpful for these cases to at least produce warning messages.

Since NeoAccess is based on btrees, debugging can involve a lot of navigating through complex data structures in the debugger. I would like to see the debugging guide in the manual expanded to include a basic overview of how to examine classes, indices and part managers.

PURCHASING

NeoLogic offers several different NeoAccess packages to meet different needs. All licenses are priced on a per-developer basis. There are no runtime licensing fees or royalties as long as the target application does not have a programmatic interface to persistent data (that is, an API or plugins) and is not a development tool.

The current prices are:

Developers	Platforms	Transferable	Price
1	Single	No	\$749
1	All	No	\$1,499
1	All	Yes	\$2,999
25	All	Yes	\$12,500

Upgrades

Upgrade prices are based on the price of the original toolkit. Minor upgrades (5.1) cost 1/6 of the original price (\$125 to \$500). Major upgrades (6.0) cost 1/3 of the original price (\$250 to \$1000). NeoLogic also offers a subscription plan at an annual cost of 2/3 of the toolkit price. Subscriptions include all upgrades for a year, plus one hour of technical support.

There is one significant flaw in the NeoAccess update policy. As developers report problems with the database engine, NeoLogic generates bug reports and bug fixes. Periodically, these are incorporated into an official release (currently 5.0.5) and the patches are posted to the NeoLogic web site. Developers can download and apply these patches,

but there is no way to obtain a "clean" bug-fix release (except for members of the NeoAccess Partners Program). Developers who bought 5.0 have already applied around 58 patches. It would be much simpler to have all the patches for a particular release encapsulated into update programs that we could download and run, or to provide entire files or functions instead of patching instructions. This may not seem like a big deal, but it's not pleasant to go through 58 of these, hoping to apply them all correctly.

```
347 Crash Resolved 5.0 - 5.0.5 All All All All
TNeoSwizzler.cpp 5.0.6
```

Applications using TNeoIDSwizzler objects crash when assigning a nil pointer to the swizzler.

```
Correcting this problem involves adding an additional check
in the TNeoIDSwizzler::operator=(pPersist *aObject) assignment
operator. The (aObject != fObject || aObject->fID != fID)
conditional at the top of the function needs to be changed to
(aObject != fObject || (aObject && aObject->fID != fID) ||
(!aObject && fID)).
```

Support

Technical support is provided by email, free for 30 days and \$120/hour thereafter. Questions about NeoAccess (asking how to accomplish something, or why a particular code sequence fails) generally receive prompt and useful answers. Some of the more difficult debugging problems (corrupted databases, for example) can't be solved without careful examination of the code. These types of questions are likely to be met with general debugging advice. The free tech support specifically excludes application-specific support; it seems reasonable to expect clients to pay if they want NeoLogic to debug their programs.

There is an email discussion group devoted to NeoAccess, where it is possible to get help from other developers who have had similar experiences. NeoLogic personnel often post responses to technical questions, and we have generally found this to be an excellent source of technical help (though this is no substitute for NeoLogic tech support when you really need it). Instructions for subscribing to the list are included on the NeoAccess CD.

NeoLogic also offers quarterly and annual support options. These all include free upgrades; NeoAccess Partners can download bug-fix releases and beta releases.

Plan	Months	Upgrades	Support	Price
Subscription	12	Yes	1 hr.	\$500
Quarterly Support	3	Yes	20 hrs.	\$1,500
Annual Support	12	Yes	80 hrs.	\$5,000
NeoAccess Partner	12	Yes	80 hrs.	\$7,500

CODE SNIPPETS: APPLICATION LEVEL

Here are examples of how to use some of the basic functionality of NeoAccess from the application level. The low-level code needed to support this is presented in the next section.

Some things to note about this code:

- `gNeoDatabase` is the NeoAccess global variable for the current database. If you are using a framework, the NeoAccess document classes ensure that `gNeoDatabase` always points to the front document's database.
- The template class `TNeoTracker` is similar to the C++ `auto_ptr` class. When it goes out of scope (either by returning from the function or throwing an exception), its destructor will delete the iterator. Without `TNeoTracker`, we would have to have a `try/catch` block in each of the functions that uses an iterator.
- The template class `TNeoSwizzler` is analogous to `TNeoTracker`, but is intended for use with persistent objects. Since persistent objects are reference-counted, application code may never use the `delete` operator on them. Swizzlers add and remove references as needed to maintain reference counts, even in the presence of exceptions. (Note that `new` and `findObject` return an object with a reference already added, so we explicitly call `unrefer` in those cases.)

Listing 1

CreateDatabase

Creating a Database

```
void CreateDatabase(const NeoFileSpec& fileSpec)
{
    // Make the database object
    CNeoDatabaseAlone* aDatabase =
        NeoNew CNeoDatabaseAlone(kFileCreator, kFileType);
    gNeoDatabase = aDatabase;

    aDatabase->SpecifyFSSpec(&fileSpec);

    // Allow the NeoAccess memory manager to write out changes to free up memory
    aDatabase->setPurgeAction(kNeoCommitBeforePurge);

    // Create the database file on disk and open it for writing
    aDatabase->create();
    aDatabase->open(NeoReadWritePerm);
}
```

Listing 2

AddPerson

Adding to the database

```
void AddPerson(const char* name, NeoDouble salary)
{
    // Create the object
    CPerson* person = NeoNew CPerson;

    // Fill in attributes of the person. For setValue, specify the tag
    // for the attribute we want to set, the type of the value we are
    // supplying, and a pointer to the actual value.
    person->setValue(pPersonName, kNeoStringType, name);
    person->setValue(pSalary, kNeoDoubleType, &salary);

    gNeoDatabase->addObject(person);
    person->unrefer();
}
```

Listing 3

FindPersonByName

Finding an object

```
CPerson* FindPersonByName(const char* name)
{
    // Make a key to select the people named 'name'.
    // pPersonName is an access tag indicating the name attribute.
    // name is the value we want that attribute to have.
    CNeoStringSelect key(pPersonName, name);

    // Find the object in the database. If more than one match, only one will be
    // returned.
    CPerson* person = (CPerson*)
        gNeoDatabase->findObject(kPersonClassID, &key);

    return person;
}
```

Listing 4

Using an Iterator

The swizzler will add a reference to whatever object is assigned to it. When a new object is assigned, the reference will be removed from the previous object.

```
void PrintPeople(CNeoIterator* iter)
{
    TNeoSwizzler<CPerson> person;

    for(person = (CPerson*) iter->currentObject();
        person != nil;
        person = (CPerson*) iter->nextObject()) {

        PrintPerson(person);
    }
}

void PrintPerson(CPerson* person)
{
    if(person == nil) {
        printf("Person is nil.\n");
    }
    else {
        // Request name and salary
        char name[kMaxNameLen];
        NeoDouble salary;
        person->getValue(pPersonName, kNeoStringType, name);
        person->getValue(pSalary, kNeoDoubleType, &salary);

        printf("%s:\t$%f\n", name, salary);
    }
}
```

Listing 5

Iterating, unordered

Iterating with a nil key will give us every object in the primary index (therefore they will be in order of object ID, not alphabetical by name).

```
void PrintPeopleUnordered()
{
    TNeoTracker<CNeoIterator> iter;
    iter = gNeoDatabase->getIterator(kPersonID, nil);
    PrintPeople(iter);
}
```

Listing 6

Iterating, ordered

We make a selection key with the tag `pPersonName`. This tells NeoAccess to use the index sorted by name. This allows us to iterate in alphabetical order.

```
void PrintPeopleOrdered()
{
    CNeoStringSelect key(pPersonName, "");
}
```



```
// Make the key match everything so we see all of the names
// (we use the key only to indicate which index to traverse).
key.setMatchAll(true);

TNeoTracker<CNeoIterator> iter;
iter = gNeoDatabase->getIterator(kPersonID, &key);
PrintPeople(iter);
}
```

Listing 7

Iterating over a subset

We make a key to select only those people with salaries of at least minSalary. If there is a salary index, NeoAccess will use it to quickly locate the set of objects we requested, and the results will be sorted by salary. Otherwise it will search all people (in the primary index) to find any that match our criteria, and the results will be sorted by object ID.

```
void PrintMoneyMakers(NeoDouble minSalary)
{
    CNeoDoubleSelect key(pSalary, minSalary);
    key.setOrder(kNeoHighOrEqual);

    TNeoTracker<CNeoIterator> iter;
    iter = gNeoDatabase->getIterator(kPersonID, &key);
    PrintPeople(iter);
}
```

IMPLEMENTING A PERSISTENT CLASS

Now that we've seen how to use NeoAccess from your application, we can look at the lower-level code that implements the CPerson class. This is a minimal class; using the advanced features of NeoAccess requires overriding more functions. The documentation describes when you must override additional functions and how to call the inherited function. Also, there are plenty of examples of how to override CNeoPersist functions built into NeoAccess.

Listing 8

Declaration of CPerson

setValue and getValue are almost identical (as are readObject and writeObject), so I have only one.

```
// Declare a unique ID for this object class.
const NeoID kPersonID = 20;

// Declare access tags used by setValue and getValue
enum {
    pPersonName = 'Name',
    pSalary = 'Slry'
};

class CPerson : public CNeoPersistNative {
public:
    // Perform any initialization specific to this class
    static void InitPersonClass();

    // Instance methods
    virtual NeoID getClassID() const { return kPersonID; }
    static CNeoPersist *New();

    // I/O Methods
    virtual long getFileLength(
        const CNeoFormat *aFormat) const;

    virtual void readObject(CNeoStream *aStream,
        const NeoTag aTag);
    virtual void writeObject(CNeoStream *aStream,
        const NeoTag aTag);
}
```

```
// Accessor methods
virtual Boolean getValue( const NeoTag aTag,
    const NeoTag aType,
    void *aValue) const;

virtual Boolean setValue( const NeoTag aTag,
    const NeoTag aType,
    const void *aValue);

private:
    // Declare the actual storage for the attributes
    CNeoString fName;
    NeoDouble fSalary;

    // Pointer to the metaclass object representing this class
    static CNeoMetaClass * NeoNear FMeta;
};
```

Listing 9

Implementation of CPerson

```
CNeoPersist* CPerson::New()
{
    // Static function to create person objects. This is registered as
    // part of the metaclass and used internally by NeoAccess.
    return NeoNew CPerson;
}

long CPerson::getFileLength(const CNeoFormat *aFormat) const
{
    // getFileLength : returns the amount of space occupied by this object in the
    // database.
    long len = NeoInherited::getFileLength(aFormat);
    len += sizeof(fName) + sizeof(fSalary);
    return len;
}

void CPerson::readObject( CNeoStream *aStream,
    const NeoTag aTag)
{
    // readObject is responsible for reading the object from the given stream.
    // By examining the stream's format object, we can support older file versions.
    NeoInherited::readObject(aStream, aTag);
    aStream->readNativeString(fName, sizeof(fName));
    fSalary = aStream->readDouble();
}

Boolean CPerson::getValue( const NeoTag aTag,
    const NeoTag aType,
    void *aValue) const
{
    // aTag indicates what attribute is being requested; aType indicates the
    // data type of aValue. We must convert the data to the requested type.

    Boolean result = TRUE;
    switch (aTag) {
        case pPersonName:
            if (aType == kNeoNativeStringType)
                *(CNeoString*) aValue = fName;
            else
                ConvertType( &fName, kNeoNativeStringType,
                    aValue, aType);
            break;

        case pSalary:
            if (aType == kNeoDoubleType)
                *(NeoDouble*) aValue = fSalary;
            else
                ConvertType(&fSalary, kNeoDoubleType, aValue, aType);
            break;
        default:
            result = NeoInherited::getValue(aTag, aType, aValue);
    }

    // Return true if getValue was successful
    return result;
}
```


Listing 10

CPerson Metaclass Object

The metaclass object registers this class with NeoAccess.

```
CNeoMetaClass NeoNear CPerson::FMeta =
    NeoNew CNeoMetaClass(
        kPersonID, // This class ID
        kNeoPersistID, // Base class ID
        "CPerson", // Class name
        // Allocator function UPP
        NeoNewGetOnePersist(CPerson::New));

void CPerson::InitPersonClass()
{
    // This function should be called at program startup to perform
    // additional initialization as needed. We use this opportunity to
    // add the name and salary indices to our metaclass object.
    // Since the indices are stock NeoAccess objects, this is all
    // we need to do to add indexing on any attribute!
    FMeta->addKey(kNeoNativeStringIndexID, pPersonName);
    FMeta->addKey(kNeoDoubleIndexID, pSalary);
}
```

CONCLUSION

NeoAccess is a very powerful object database management system well-suited to a wide variety of data storage applications. After overcoming the steep learning curve, developers should find the product relatively easy to use. The documentation is helpful, but mastering the more complex features of NeoAccess requires experience with the database engine.

Pricing and upgrade policies are mostly reasonable. The lack of runtime licensing fees for most applications make this an attractive option for commercial software products. NeoAccess is used in the latest versions of Netscape Communicator, America Online 3.0 and NetObjects Fusion.

For more information, check out the NeoAccess Technical Overview. The overview, plus source code and executables for the demo programs, are available on the NeoLogic web site.

URLs

- NeoLogic, Inc. Web Site – <<http://www.neologic.com>>.
- NeoAccess Technical Overview – <<http://www.neologic.com/archovrw.html>>.
- NeoForum Mailing List. Subscribe or view archived messages at – <<http://clio.lyris.net/cgi-bin/lyris.pl?site=neologic&page=list&list=neoforum>>.

T

Want to subscribe to MacTech Magazine? Contact
<<mailto:orders@devdepot.com>>, call 800-MACDEV-1 or 805-494-9797, or fax 805-494-9798.

RHAPSODY
READY
POWER

NOW

THERE'S
NO REASON
TO WAIT

POWER YOUR RHAPSODY APPLICATIONS WITH OPENBASE — the high-performance SQL database server for Rhapsody! Tap into proven technology, it's ready now!

RELEASE THE POWER — make applications that are scalable, multi-user and fast with OpenBase.

- Aggressive Multi-threading
- Text Searching
- Change Notification


- Row Level Locking
- Variable Record Length Technology

THE POWER TO BUILD — make building applications easy with the full line of graphical tools, native EOF adaptors, and Rhapsody source code examples that come with OpenBase.

THE POWER IS YOURS

Order now and take advantage of our

FREE SINGLE-USER RUNTIMES



SPECIAL OFFER
5 CONNECTIONS
JUST \$495!

OPENBASE®
INTERNATIONAL

Save Characters

http://www.openbase.com

by Gene Backlin

OpenBase

A Solid Database Framework for Rhapsody

INTRODUCTION

OpenBase is a solid database framework which will address your data handling requirements for Rhapsody. Like Rhapsody, OpenBase's foundation is with NeXTSTEP. It has evolved through the years to provide a mature environment for stand alone users as well as over distributed networks. For the developer, OpenBase has a rich set of application APIs that incorporate the C and Objective-C languages. Regardless of developers programming background Mac OS or NeXTSTEP/OpenStep, the OpenBase API framework allows quick development of full scale database applications.

OVERVIEW

This article will illustrate:

- Steps to build SimpleTool, an application that queries a "Movie" database, show just how simple is writing an OpenBase database application.
- Help Desk, an application using OpenBaseAdvancedAPI to address multi-tiered database interaction over local area networks.
- OpenBase Manager, OpenBase's data management and interactive tool.

SIMPLETOOL

SimpleTool demonstrates interaction with a relational database interaction without using the tedious programming overhead common with databases.

Using C or Objective-C is the simplest way to access OpenBase. SimpleTool will retrieve from the database the movies and the revenue from the producing studios. Listing 1 illustrates the OpenBase API framework. A discussion follows.

Listing 1: SimpleTool_main.m

```
#import <Foundation/Foundation.h>
#import <OpenBaseAPI/OpenBase.h>

int main (int argc, const char *argv[])
{
    NSAutoreleasePool *pool = [[NSAutoreleasePool alloc] init];
    int returnCode;
    OpenBase *connection = ob_newConnection();

    //variables to hold values
    char movieTitle[256];
    char studioName[256];
    long revenue;

    if (!ob_connectToDatabase
        (connection, "Movie", "", "", "", &returnCode))
    {
        printf("%s\n", ob_connectErrorMessage(connection));
        return -1;
    }

    ob_makeCommand(connection, "select t0.TITLE, t0.REVENUE,
        t1.NAME from MOVIE t0, STUDIO t1 where
        t0.STUDIO_ID = t1.STUDIO_ID order by t0.REVENUE DESC");

    if (!ob_executeCommand(connection))
    {
        printf("ERROR - %s\n", ob_serverMessage(connection));
        ob_invalidate(connection);
        return -1;
    }

    ob_bindString(connection, movieTitle);
    ob_bindLong(connection, &revenue);
    ob_bindString(connection, studioName);
```

Gene Backlin <gbacklin@MariZack.com> has been programming since 1978, and is owner and principal consultant of MariZack Consulting, formed in 1991 with one purpose — to help. He has been helping clients such as IBM, McDonnell Douglas, Waste Management Inc., the U.S. Environmental Protection Agency, AT&T, Ameritech, Discover Card, Rockwell International, Bank of America and Nations Bank. He also helps local universities in the area of education and is author of the book "Developing NeXTSTEP Applications" ISBN 0-672-30658-1 published by SAM's Publishing.

Spotlight Memory



NEW Debugger

Spotlight is the first automatic memory debugger for the Macintosh. Instantly detect invalid memory accesses, memory leaks, bad toolbox parameters, stack overwrites, memory relocation problems, and more. Spotlight can find problems in seconds that might take you a week to fix by hand.

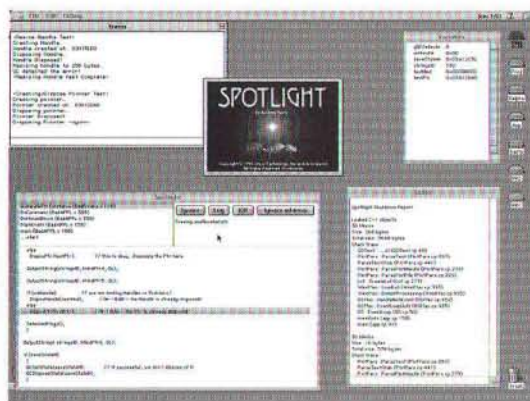
Spotlight uses your XSYM file to automatically patch your PowerPC native application. When Spotlight detects an error it brings up the exact source code line where the error occurred along with stack crawls, memory views, and program variable usage information.

No need to change your source code. No need to recompile. No learning curve whatsoever.

Spotlight's Object Code Replacement instrumentation inspects each PowerPC processor memory read and write instruction in your code as it runs. Spotlight detects faulty memory reads and writes that occur between blocks, in released blocks, across multiple blocks, and outside of your heap.

Features

- Stops your program on exact source code line where a bad memory write occurs
- Complete leak reporting with source code stack trace showing point of allocation
- Fully validates parameters to over 400 Mac toolbox API calls.
- Works on all heap allocations Mac Toolbox, C 'malloc', C++ new and delete.
- Supports any language that produces an MPW or Metrowerks XSYM file.
- Full logging and ignore capabilities
- API for fine tuned control
- Purchase includes one year subscription to downloadable updates
- 30 day money back guarantee
- Available for secure electronic purchase now at our web site



Requirements

Macintosh or Mac OS-compatible computer with a PPC 601 or greater processor; RAM requirements vary depending on target application size; 2 MB hard disk space; System 7.5 or later; Metrowerks or MPW compatible XSYM symbolic file

Download free demo at <http://www.onyx-tech.com/>

\$199.00

TEL (941) 795-7801 **E-MAIL** sales@onyx-tech.com
FAX (941) 795-5901 **URL** <http://www.onyx-tech.com/>

```
while (ob_nextRow(connection))
{
    printf("%s made $%ld for %s.\n", movieTitle, revenue,
          studioName);
}

ob_invalidate(connection);
[pool release];
exit(0);
return 0;
}
```

Main begins by establishing a connection to the database, if a connection was not made, print the offending message returned from the connection object and exit. Using the `ob_connectToDatabase()` function, establish a connection to the database with the database name, hostname, logon id and password. Note the argument `""`. The database connection is made directly to the local machine on a standalone computer, but will poll all hosts on a local area network. This is an example of OpenBase's scalability.

```
int returnCode;
OpenBase *connection = ob_newConnection();
//...
if (!ob_connectToDatabase(connection, "Movie", "", "", "",
    &returnCode))
{
    printf("%s\n", ob_connectErrorMessage(connection));
    return -1;
}
```

After a successful connection has been established, the `ob_makeCommand()` function is used to send SQL statements. The

TITLE and REVENUE data columns from the MOVIE table as well as the associated studio NAME from the STUDIO table are retrieved. The SQL statements are now buffered for later execution by the database server.

```
ob_makeCommand(connection, "select t0.TITLE, t0.REVENUE,
t1.NAME from MOVIE t0, STUDIO t1 where t0.STUDIO_ID =
t1.STUDIO_ID order by t0.REVENUE DESC");
```

The `ob_executeCommand()` passes the buffered SQL statements to the database server and returns TRUE for successful and FALSE for failed execution.

```
if (!ob_executeCommand(connection))
{
    printf("ERROR - %s\n", ob_serverMessage(connection));
    ob_invalidate(connection);
    return -1;
}
```

The `ob_bindString()` and `ob_bindLong()` functions, bind the resulting data columns from the database, to the receiving program variables. SimpleTool binds the variables `movieTitle`, `revenue` and `studioName` respective to the order of the initial SELECT statement.

```
ob_bindString(connection, movieTitle);
ob_bindLong(connection, &revenue);
ob_bindString(connection, studioName);
```


Have You Made a Visit to the New MacTech® Web Site?

<http://www.mactech.com>

Here are some highlights:

▶ Macintosh Development News

Keep yourself informed. The site has all the current breaking news in the development community and even archives past news pieces.

▶ Search the Site

Search like never before. Search and find that article, news piece or source code you're looking for via the new Phantom search engine courtesy of Maxum.

▶ Article Archives

And you can have it all. These archives have thousands of pages of content in them from the histories of MacTech, MacTutor, develop, and FrameWorks magazines.

▶ develop

develop. Need we say more. The entire history of develop, Apple's award winning technical journal. That is 29 issues from 1990 - 1997.

The screenshot shows the MacTech Magazine website as it appeared in 1997. The browser window is titled "Netscape: MacTech Magazine" and the address bar shows "http://www.mactech.com/". The website features the MacTech logo at the top, followed by a banner for "Developer DEPOT" with the text "Hundreds of products and the lowest GUARANTEED prices!". Below the banner, there are three main sections: "Click Below" with a list of links, "MacTech CD-ROM" with a description of the CD-ROM content, and "Macintosh Development News" with a list of recent news items. At the bottom, there is a "Search News..." field and a "Search" button. The website also includes a "Current Issue" section and a "Need an answer on a programming question?" section.

Click Below

- ▶ About MacTech
- ▶ Home Page
- ▶ Subscribe to MacTech!
- ▶ MacTech CD
- ▶ MacTech Japan
- ▶ Get a free copy of MacTech
- ▶ MacDev-1
- ▶ Developer News
- ▶ Search The Site
- ▶ Getting Started
- ▶ MacTech Online
- ▶ Programmer's Challenge
- ▶ Job Postings
- ▶ Article Archives
- ▶ Source Code/FTP Site
- ▶ Writer's Kit
- ▶ Contact the Editors
- ▶ Editorial Calendar
- ▶ Advertising
- ▶ Developer Depot Worldwide
- ▶ Developer Depot Japan
- ▶ THINK Reference
- ▶ Developer Central
- ▶ MacHack
- ▶ How our net is done
- ▶ Legal/Disclaimers
- ▶ Webmaster Feedback

MacTech CD-ROM

The MacTech CD-ROM now includes every article available from every major Mac developer publication all in THINK Reference format.

Macintosh Development News

- 12/02/97 PR : Metrowerks Announces First Quarter 1998 Revenues of US\$6.2 Million
- 12/02/97 PR : WebSTAR PowerKey Pro Tickler
- 12/02/97 PR : The Association of Macintosh Trainers
- 12/01/97 PR : Apple Announces WebObjects 3.5
- 12/01/97 PR : Apple Delivers Public QuickTime 3.0 Developer Preview Release
- 12/01/97 PR : Roaster Technologies Licenses ObjectSpace Voyager As Primary Java Interface For Distributed Computing
- 11/25/97 PR : Apple, BLACKSMITH Offer Classes On Creating Dynamic Web Applications
- 11/25/97 PR : Bridge Allows Mac Webmasters to Easily Upgrade from Command Tags to XML
- 11/25/97 PR : Joy Introductory Offer Expires This Week
- 11/25/97 PR : Pyromaniac Pro
- 11/24/97 PR : SpotCheck, the Program Editor That Knows Java
- 11/24/97 PR : SiteWarrior Wages War On Web Site Complexity
- 11/24/97 PR : Objective-Everything Release 5
- 11/24/97 PR : Maxum Development Announces Release Of NetClock 2.5 Upgrade
- 11/24/97 PR : Build Counter Tool for Tracking Number of Builds
- 11/20/97 PR : Object Plant Version 1.4.4

News prior to 11/20/97.

Search News...

Advanced Search...

Need an answer on a programming question? Check out our complete archives -- thousands of pages -- including MacTech, MacTutor, develop and FrameWorks.

ob_nextRow() increments through the result rows and retrieves the data. FALSE is returned when all data is processed.

```
while (ob_nextRow(connection))
{
    printf("%s made $%ld for %s.\n",movieTitle, revenue,
        studioName);
}
```

Main ends with a call to terminate the connection to the database server.

```
ob_invalidate(connection);
```

HELP DESK

Help Desk, addresses multi-tiered database requirements by directly connecting to the user interface through the OpenBaseAdvancedAPI as seen in **Figure 1**.

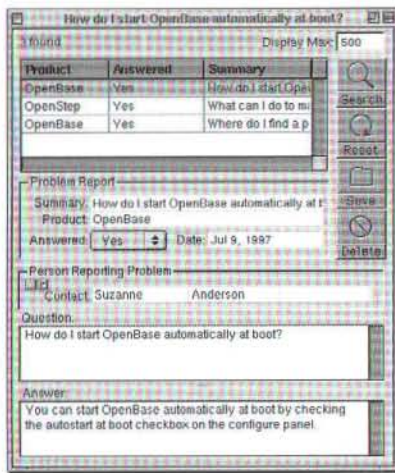


Figure 1. Help Desk Manager.

Designing the Interface

The Apple supplied tool Interface Builder is used to design the user interface. Information on Interface Builder is detailed at <http://devworld.apple.com/techinfo/techdocs/rhapsody/apple.com>. See **Figure 2** for a screen shot of a connection being performed using Interface Builder.



Figure 2. Making the Interface Builder Object Connection.

Managing the Interface and Database Relationships

Two FormManagers and one ReadTableManager are created. The first FormManager lists customer questions and the other displays who asked the question. The ReadTableManager manages a picklist of information.

```
tableManager = [[ReadTableManager alloc] init];
formManager = [[FormManager alloc] init];
contactsFormManager = [[FormManager alloc] init];
```

To set the key attribute and table, each object must be initialized.

```
[tableManager setKeyAttribute:@"support._rowid"];
[tableManager setTableName:@"support"];
```

```
[formManager setKeyAttribute:@"support._rowid"];
[formManager setTableName:@"support"];
```

```
[contactsFormManager setKeyAttribute:@"contacts._rowid"];
[contactsFormManager setTableName:@"contacts"];
```

Connect the interface objects; tableManager, formManager and contactsFormManager to the database.

```
[tableManager setConnection:connection];
[formManager setConnection:connection];
[contactsFormManager setConnection:connection];
```

Set specific query ordering. In this example "ASC" is ascending.

```
[tableManager setOrderBy:@" ORDER BY
support.shortQuestion ASC"];
```

Bind the screen objects to the database columns.

```
#define FIELD(outlet) [OutletManager newForOutlet:outlet]
#define TEXT(outlet) [OutletTextManager newForOutlet:outlet]
#define POPUP(outlet) [OutletPopUpManager \
    newForOutlet:outlet]
```

```
[formManager addOutlet:FIELD(shortQuestion)
    withColumnName:@"support.shortQuestion"];
[formManager addOutlet:FIELD(product)
    withColumnName:@"support.product"];
[formManager addOutlet:TEXT(textQuestion)
    withColumnName:@"support.question"];
[formManager addOutlet:TEXT(textAnswer)
    withColumnName:@"support.answer"];
[formManager addOutlet:FIELD(dateCreated)
    withColumnName:@"support.dateCreated"];
[formManager addOutlet:POPUP(answered)
    withColumnName:@"support.answered"];
```

Set up the Contacts Form to be a target of the FormManager.

```
[contactsFormManager addOutlet:FIELD(firstname)
    withColumnName:@"contacts.firstname"];
[contactsFormManager addOutlet:FIELD(lastname)
    withColumnName:@"contacts.lastname"];
[contactsFormManager addOutlet:FIELD(email)
    withColumnName:@"contacts.email"];
```

Add each column to initialize the table.

```
[tableManager addColumn:@"support.product"
    title:@"Product"];
[tableManager addColumn:@"support.answered"
    title:@"Answered"];
[tableManager addColumn:@"support.shortQuestion"
    title:@"Summary"];
```


Initialize the windowTableView object, to display the query results.

```
[tableManager setTableView:windowTableView];
```

Establishing Database Relationships

The formManager will display related details when a selection is made in the tableManager. To accomplish this, a target relationship must be made between the tableManager and formManager.

```
[tableManager addTarget:formManager  
withValue:@"support._rowid"];
```

The contactsFormManager displays the contacts through the contacts_id key.

```
[formManager addOutlet:contactsFormManager  
withColumnName:@"support.contacts_id"];
```

The formManager sets the SQL "WHERE" constraints as well as subqueries.

```
[tableManager fetchData:[formManager whereConstraints]];
```

Insulation from Direct SQL

OpenBase's OpenBaseAdvancedAPI, completely insulates you from SQL commands like SEARCH, RESET, SAVE and DELETE, by the following methods.

```
- (void)findAction:sender  
{  
    [tableManager fetchData:[formManager whereConstraints]];  
}  
  
- (void)resetAction:sender  
{  
    [tableManager resetAction:self];  
}  
  
- (void)saveAction:sender  
{  
    [formManager saveChanges];  
}  
  
- (void)deleteAction:sender  
{  
    [tableManager deleteAction:self];  
}
```

THE OPENBASE MANAGER

In addition to the developer API frameworks, OpenBase like Rhapsody, designed graphical tools to simplify tasks. OpenBase Manager simplifies the following:

- Managing Database servers across local area networks
- Viewing Databases
- Editing Database schemas
- Managing Database security

OpenBase Interactive Database Toolset

The following screen shots display OpenBase's rich set of tools.

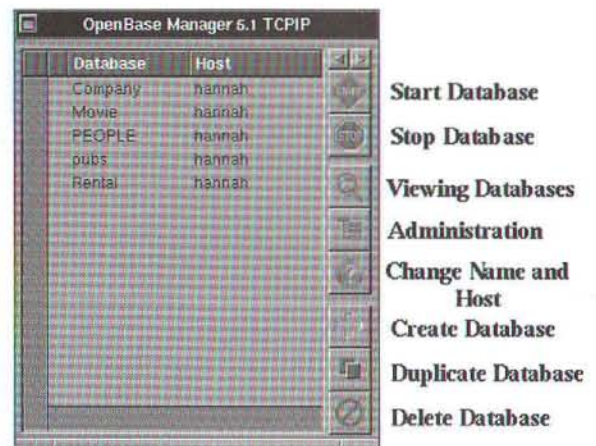


Figure 3. OpenBase Database Manager.

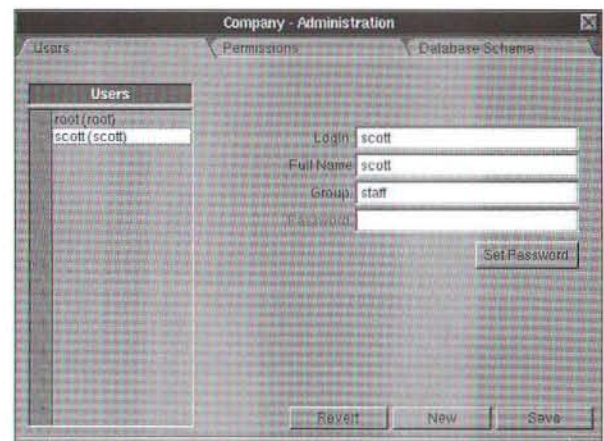


Figure 4. User Administration.

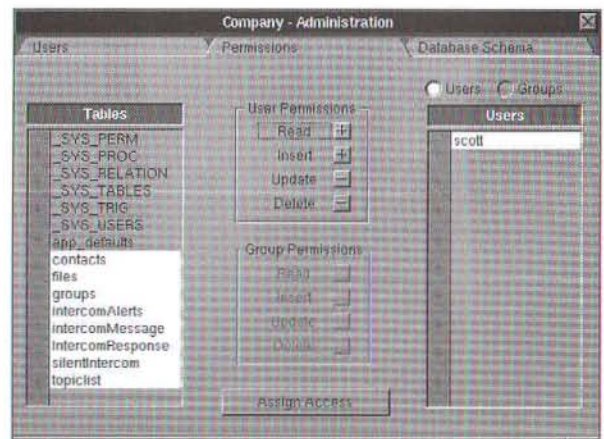


Figure 5. Permission Administration.

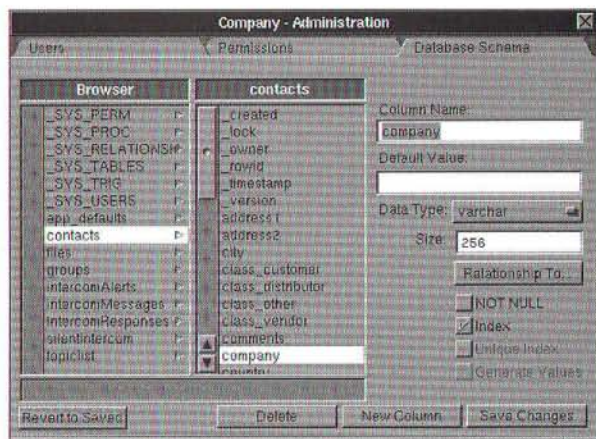


Figure 6. Database Table Administration.

USER COMMENTS

Sirius Connections, a leading provider of internet services for the San Francisco area, uses OpenBase for billing and maintaining historical records on 15,000 customers. "Our whole operation is built on OpenBase technology," says Andreas Glocker, CEO of Sirius Connections, "Automating our business on OpenBase has made all the difference. It has given us the competitive advantage."

"One of our programmers wrote a system using the OpenBase API in less than a day. Doing the same thing using Oracle OCI's took more than three," says Kevin Ford, President of ComputerActive located in Ontario Canada, "OpenBase demonstrates a level of quality and robustness rarely seen in the software world."

Robert L. Peek, founder of the Peek Financial Group, says, "We have adopted OpenBase as an enterprise wide solution for our firm. We have found it to be an industrial strength database with excellent support."

CONTACT INFORMATION

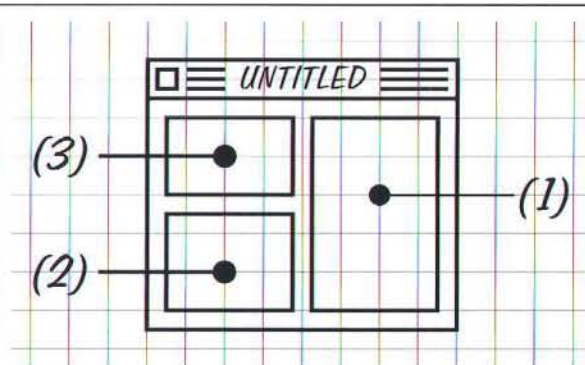
OpenBase supports ODBC for Mac OS and Windows and has a native JDBC driver. For further information about OpenBase and how you can get a FREE single-user runtime, you can contact:

OpenBase International, Ltd
58 Greenfield Road
Francestown, NH 03043
Tel: 603-547-8404 — Fax: 603-547-2423
e-mail: <info@openbase.com>
internet: <http://www.openbase.com>



Stop reading hex.

Save time with
Quadrivio General Edit,
the fast way to edit data files



- 1 Define your file format**
 - Easy, familiar C-like statements.
 - Complex formats handled by powerful syntax.
 - Input is checked and compiled *as you type*.
- 2 View, edit, or analyze your data.**
 - Edit data *directly* in window.
 - Automatically check values, at your option.
 - File can be larger than RAM.
- 3 Multiple uses:**
 - Modify Finder Info and file parameters.
 - Also edit data in application heaps, memory handles, and parameter RAM.
 - View entire disk volumes.

Quadrivio General Edit ADVANTAGES

- ✓ **Quick insight** into data structure contents.
- ✓ **Easier** than studying MPW's dumpfile output.
- ✓ **Faster debugging** with clear display of data.
- ✓ **Saves coding time** with direct data editing.

NEW On-line version \$195

Start using General Edit just 30 minutes from now.
Visit our web site. See, Learn, and Download.

<http://www.quadrivio.com>

Quadrivio Corporation info@quadrivio.com
1563 Solano Avenue #360 Berkeley, CA 94707 (510)524-3246

by Dave Mark ©1997 by Metrowerks, Inc., all rights reserved.

Neil Ticktin, MacTech Publisher

If you are an old-time Mac programmer like me, chances are you remember the original Macintosh technical journal, then called *MacTutor*. Remember the MouseHole? Aztec C? Light speed Pascal? *MacTutor* filled in the technical holes for a lot of aspiring Macintosh developers. If I recall correctly, it was at August *MacWorld* when I first met Neil Ticktin. Word was that *MacTutor* was having some difficulties, and Neil was going to take over the magazine to see if he could make a go of it. We talked about my doing a regular column for beginners, starting with the January 1992 issue of *MacTutor*.

Now it is more than *six* years later. The magazine changed names to *MacTech*, grew a new look and feel, has been through a number of personnel changes from top to bottom, added a mail order catalog, absorbed Apple's *develop* magazine, and helped spawn *MacWorld's* Developer Central. It has been a wild ride.

At this past January *MacWorld*, Neil and I had some long talks about the state of Mac development and the direction Apple was taking the Mac OS. Make no mistake here, Neil is no hard-core developer. But he has a unique perspective on the state of the Mac that I thought you might find interesting and valuable. So, without further ado...

Dave: How has Apple's shrinking market share affected MacTech Magazine?

Neil: The interesting thing about MacTech Magazine is that it targets developers, not users, and, as a result, tends to be a good future indicator. For example, MacTech had less readers in 1995-6 which preceded Apple's decline. MacTech, over the past year/year and half, has actually had growing readership. And, in addition to that, we launched MacTech Japan which has done very well. So, we're pleased with how things are going — and we think that 1998 will be a much better year for developers than 1997 was.

The interesting thing about market share is that people are generally talking about sales of computers, not installed base. To most developers, installed base is what is important. And, if you look at Macs in current use as a percentage of all personal computers, you probably would see market share closer to 10%. And the percentage is probably larger if you talk about active Mac users compared to the overall number of active personal computer users. Now, as a developer, do you care how many machines Apple sells? Or do you really care about how many customers can buy your product?

The disparity caused by sales market share and installed base market share comes down to two things. First, in most cases, Macs are better built and will last longer than their WinTel counterpart. Second, Apple has done a pretty good job of backwards compatibility which translates into preserving their customers' investments. This may not be the most profitable thing for Apple, but it does wonders for the customers. It's unfortunate, but OS and hardware vendors make more money if they build products that become obsolete quickly. This is not what Apple or the Macintosh are about.

Dave: I was surprised with the lack of Rhapsody presence at January MacWorld. While there were some Rhapsody apps being shown at Developer Central, there was no mention of Rhapsody at all in Steve Job's keynote. Where do you see Rhapsody going from here?

Neil: For a while there, Apple was into selling the future. Apple seems to have learned the lessons of the past, and today, Apple focuses on selling what it can deliver right now. In addition, with Mac OS 8, Apple realized that the Mac OS still has legs. These two things, I believe, are the main reasons behind Apple's down playing of Rhapsody (or maybe it was just up-playing the Mac OS) at Macworld San Francisco.

Also, I think that you will see some blurring of the benefits of future Mac OS's in comparison with Rhapsody. Apple has eluded to bringing more modern OS features to the Mac OS — whether this comes from Copland work or Rhapsody work being integrated into the Mac OS, we've not seen yet — but it looks to be coming.

As a Mac OS or even OPENSTEP developer, you should watch the Mac OS efforts. It may be the case that the feature you really needed gets incorporated into the Mac OS — and today, that has a lot bigger set of customers.

Rhapsody is coming along, and there is developer support of it. There are a number of tools vendors already supporting Rhapsody: Metrowerks, AAA+, SNAP, TipTop, OpenBase, etc... And, that's for an OS that has no real market today. When Rhapsody is ready, I would expect Apple to take the approach of easing it in with as little cannibalization of the Mac OS as possible. Hopefully, they will be able to expand the market for developers, not just replace what we already have. That means that you'll see special attention paid first to those users who have the greatest need for features only found in Rhapsody.

As a Rhapsody developer, you should pay attention to what Apple is targeting first. If you don't match your efforts to where the potential customers will be, you will probably have a problem.

Dave: How do you see the economics of selling into the Mac OS market in 98?

Neil: Interesting thing about Mac OS 8 — it's been selling well. In fact, according to a November 97 SoftTrends report, Apple took the top market share position in August 97 with 35.1% of units sold, compared to Windows 95 at 27.8%. This was the first time that any Mac title attained the number one ranking. Read this again — the Mac OS outsold Windows 95. Now yes, this is a brand new OS vs. a 2 year old OS, but it is



SERIAL EXPANSION BOARDS PARALLEL EXPANSION BOARDS

- 2/4 PORT SCSI DEVICES
- 2/4 PORT NUBUS SERIAL BOARDS
- 4 PORT PCI SERIAL BOARDS
- NUBUS & PCI PARALLEL BOARDS

BOARDS WORK WITH SERIAL SCANNERS, GRAPHIC TABLETS, LABEL WRITERS, MODEMS, ISDN MODEMS, AND OTHER COM TOOLBOX SAVVY HARDWARE AND SOFTWARE PRODUCTS.

TO ORDER CALL: 800-367-8465

7509 Connelley Drive • Suite D • Hanover, MD 21076

www.creative-solutions-inc.com
creatsolns@aol.com



MAC OS IS A REGISTERED TRADEMARK
OF ITS RESPECTIVE OWNER.

a good indicator that the Mac is a lot healthier than the media would lead you to believe.

Another interesting point to think about is that Microsoft operating systems certainly have some 90% of the installed base. But realize that this is divided between DOS, Windows 3.1, Windows 95 and Windows NT. There are some pieces of software that run on both 95 and NT, but a lot of software has separate versions for each of these OS's, or maybe just doesn't run on them. The NT market share is far less than the Mac OS. And, when I saw numbers last summer, only 19% of the market had upgraded to Windows 95 — nearly two years after it's release, and it's market share was not even double the Mac OS installed based at the time.

A couple of years ago, we were looking at the business prospects for Mac developers. What we found is that yes, obviously, the Mac has a smaller market size. But, interestingly, when you look at the Windows market, factor in the disproportionately higher marketing, tech support, QA, development, and other costs — the return on investment was higher on the Mac than it was under Windows. Today, that return has evened out, but it still looks like the Mac OS development is close to the return on investment that Windows development has.

And, all of this is during a time that has been hard for Apple. As of this writing, 1998 is looking to be a much stronger year for Apple and the Mac. Apple just turned it's first profit in a while (and remember, due to prior losses, Apple has tax free profits for a long time to come). A lot of shakeout has already happened, and some developers (like Casady & Greene Inc.) reported that this last Macworld was one of their best shows ever! Now is the time of opportunity for Mac developers as the market starts to grow and prosper.

Don't get me wrong, I don't want to blindly support Apple or Macintosh — we just do things that make sense around here, and we make up our own minds. Win95 has certainly a lot more units out there. As a developer though, you need to make decisions based on what you know about what you can do — not what some reporter, finance person, or Wintel marketing person tells you. I can tell you from experience, there are a lot of people out there that are succeeding on the Macintosh, and wouldn't give it up.

Dave: What kind of changes do you see coming down the pike with Developer Central?

Neil: As you know, MacTech and Apple created Developer Central together in January 1995. Today, now in our 4th year, Developer Central continues to push the envelope — demonstrating not only tools for the Mac OS, but Rhapsody as well. You may have noticed that in Macworld Expo's roundup, Developer Central was listed as one of the top 3 successes of the show with about 45 technologies showcased, it represented some 10% of the exhibiting companies at the show.

The cool thing about Developer Central this past January is that we are seeing some really neat things pushing the envelope. Take products like Resorcerer, which is getting support added disassembling code. Or, AAA+'s Joy which can be used to really learn a lot about Rhapsody development. This is just to name a couple of cool tools — there are a lot more.

Dave: I thought it was interesting that you were asked to present at the Eddy awards at January MacWorld. What do you make of that?

Neil: Over the past couple of years, the Mac community has come to realize how important developers are. This year, Macworld and MacTech magazines joined together to really give the developer community the level of support it deserved. Which tools received nomination status, let alone the winner status, was a hard decision. There were so many different products that are worthy. We're very proud of the finalists and the winners. We're very pleased that *Macworld* Magazine agrees that developers are critical and that we've joined together to reward them.

Dave: What options do you think are available to developers who are new to the Macintosh? How about long-time Mac developers?

Neil: It all depends on what your goals are, what your project is about, and what market or functionality segment you are in. I know that you've heard this a million times before, but just because the Windows market is bigger, doesn't mean that it's better. For example, if you are in an education or creative market, these are extremely strong for the Mac. Likewise, if you don't want to build a large organization for marketing, QA, testing, tech support, etc... you may be very satisfied with a smaller Mac market with greater profit margins, less competition, and in general, a lot less hassle to bring your product to market.

What we hear today from long time Mac developers is that during the past year or so, they've forced themselves to look around at options beyond the Mac OS. Most have looked first at Windows NT or Java.

What they've concluded in many cases is that even with a lot of progress behind NT, it's still Windows. It still is not a pleasure to use. You still have to reinstall software, fiddle with drivers, spend hours installing a hard drive, etc... It still is a poor interface by comparison to the Mac. And the little voice inside their head, the artist side of them, says "I don't like this as much as the Mac. Do I really want to work on something I don't like?"

Java is another beast entirely. There are a lot of really cool things behind Java. And, the notion of cross platform development is appealing. Unfortunately, even with a lot of progress here as well, it's not quite ready for "prime time". And, with all of the power struggles over Java, people are concerned. So, developers are using Java — and are keeping an eye out for where's applicable.

So, with those experiences, developers have been saying "Hmmm... maybe the Mac OS is not such a problem. Maybe I just like working on it and the business proposition is ok and there is opportunities there. And, if Apple is able to deliver on Rhapsody, I can move my development there and then target Rhapsody, Rhapsody on Intel, Mac OS, Windows 95 and Windows NT all with one set of sources." So, Mac developers are looking for a high quality environment to work in, and be successful at it — a combination of Mac OS, Java and Rhapsody seem to be the key for most.

Dave: Can you contrast the US market with the Japanese market and offer some suggestions for developers who want to move into the Japanese market?

Neil: First, let me say that Japan is a wonderful market. It's a wide open Mac market that you need to explore. It is a long

term play, and you need to carefully look at your options. Don't expect to fly in, find a partner, fly home and have the deal done in a week. It usually takes months to put something in place.

Market share for Macintosh in Japan is roughly double that of the U.S. for a country that has almost half as many people as the U.S. population. Apple sells about 25% of its units into Japan. And, unlike the problems in the Japanese PC market due to an economic downturn, the Mac market has been much stronger.

Japan offers a completely different type of economy than we are used to in the U.S. For example, a copy of MacTech Magazine goes for US\$5.85 on the newsstand. In Japan, MacTech Japan sells for ¥2500-¥2800 — roughly equal to about US\$20 or more. And, to give you an idea of the developer market, MacTech Japan has consistently sold well. Similarly, companies like Metrowerks Japan have done exceptionally well in this market.

The first suggestion that I can offer is to GO to Japan. You need to learn about the culture, and experience something like a Macworld. Look at magazines on the newsstand, and visit shopping areas like Akihabara [we need to check the spelling on this]. You will almost definitely need a Japanese partner to succeed. You won't be able to penetrate the market without someone having a native understanding of what is going on. You should talk to fellow developers and see who they have used. A personal recommendation for a local partner is an excellent way to go.

Dave: Any advice you'd like to offer to the Mac developer on the edge, considering moving to the Windows market?

Neil: First of all, figure out which platform is the best way to deliver your product. If you are in education or creative, the Mac is the definite winner. But, if you are doing legal or accounting software, Windows is probably your better bet. Worry about creating great products for whatever platform and then sell the benefits of that product. Remember, on the internet, no one knows you're a Mac. So, if you have something they definitely need — who cares what machine it runs on? Or use the Mac as your place to do your primary development, and then deliver on multiple platforms if that makes sense for you. It all comes down to what you want, and what you know about where your product can succeed. [1]

Want to suggest an article for the magazine? Send your suggestion to
[<mailto:editorial@mactech.com>](mailto:editorial@mactech.com)

Build great applications... Better Cheaper Faster!

Great Good'nuf Tools Plus LIBRARIES + FRAMEWORK

Make cool apps while others are still reading their manuals. Professionally polished. Wickedly fast. Delightfully efficient.

"...the routines are more compact and faster than anything you might write... Every element of Tools Plus is useful... a bargain compared with coding these routines yourself."



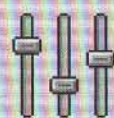
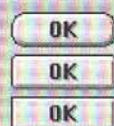
You build the interface.

Tools Plus provides the infrastructure.

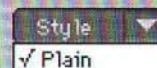
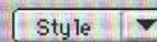
It makes all your pieces work together as an application. With only a few hundred routines, Tools Plus thins your code by tens of thousands of lines. You see results sooner. Changes are a snap.

"All in all, it's an incredibly rich collection of tools... If you are interested in developing applications that have 'quality' written all over them, then Tools Plus is for you." **MacTech MAGAZINE**

Yes, Tools Plus has it!



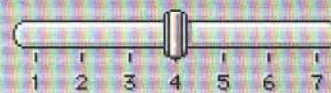
- Create any element using a single routine
- Everything works as soon as you create it
- Automates all standard GUI elements
- Windows, Tool bar and floating palettes
- Buttons (all kinds, flat and 3D)
- Scroll bars (speed control, live scrolling)
- World-class custom controls
- Fields (w/scroll bars, filters, auto-editing)
- Picture buttons (the best anywhere)
- List boxes
- CDEF and LDEF automation
- Cursors (color, animated, auto-change)
- Menus (pull-down, hierarchical, pop-up)
- Edit menu (undo/redo, automatic editing)
- Panels (3D, group boxes, lines, panes)
- 3D titles (raised or inset)
- Extended multi-monitor and color support
- Clipboard automation
- Dynamic alerts (no resources required)
- Event processing automation
- Over 500K of custom fonts, icons, cursors, and other useful resources



Plus thousands more exciting features and services!

Results:

- ☒ Radical
- ☐ Slick
- ☒ Ravin'



FREE

SuperCDEFs™ \$89 value

- ✓ professionally designed and crafted controls
- ✓ dozens of 3D & flat buttons, tabs, sliders

Tools Plus for	
Symantec (THINK) C/C++ (68K)	\$149
THINK Pascal (68K)	\$149
THINK C/C++ & THINK Pascal	\$199
CodeWarrior Bronze (68K)	\$199
CodeWarrior Gold (68K and PowerPC)	\$249

(We accept VISA and Amex only. Add \$10 for shipping.)
 *Call for Academic pricing.

Water's Edge Software
 2441 Lakeshore Road West, Box 70022,
 Oakville, Ontario Canada L6L 6M9

Orders and Enquiries:
 Phone: (416) 219-5628
 Fax: (905) 847-1638

WaterEdgeSW@aol.com

Free Evaluation Kit:

Available at our web site
<http://www.interlog.com/~wateredge>



by Mark Holtz

Using Apple Information Access Toolkit

*Apple's new indexing
technology provides
powerful ways to search
and retrieve your data,
regardless of how it's stored*

EXACTLY WHAT IS AIAT, AND WHAT CAN IT DO FOR YOU?

The Apple Information Access Toolkit (AIAT) is a library of routines designed to distill, index, and query collections of textual data. The elegance of the technology stems from its complete independence from the actual data source it is indexing. It can be used in a fully-interfaced, stand-alone application, or in a memory-constrained plug-in. You can feed it everything from your next-generation database to a catalog of your Compact Disc collection. Best of all, rumor has it that AIAT will be a part of Rhapsody, so your hard work now will pay off in the future.

Sounds great! How do I use it?

If you haven't guessed by now, you're going to have to get your hands dirty and write some code. AIAT provides a set of C++ objects that can be sub-classed to provide the desired functionality. Strange runtime quirks notwithstanding, the overall process is actually easier than it may first appear. In a recent development

project, I created an AIAT-based plug-in that indexed and queried data from a 3rd party database in less than two days. This article will focus on that development project and some of the lessons I learned. The most important lesson was that AIAT will do a lot of the work for you, but you have to tell it exactly what you want.

AIAT: A TECHNOLOGY OVERVIEW

Upon unwrapping the AIAT package, you will find a nice, clean set of components to speed you on your way. Documentation is provided in Adobe PDF format, and is relatively comprehensive in scope, but occasionally lacks the necessary depth to help you fully understand a particular object. Next, a set of 68K and PowerPC libraries for Metrowerks' CodeWarrior 11 and Pro 1. Where there are libraries, there are also headers. AIAT's headers comprise what I consider to be the missing portions of the documentation. Not only are they slightly more current than the documentation, but also provide the necessary implementation details one needs to avoid MacsBug. The C++ classes in AIAT are robust, but you only need to deal with a small number of methods to create a working application. Finally, there are two example applications, one that indexes files in a folder and another that lets you query against the generated indices. They are fairly complete in their coverage of basic AIAT concepts from a client perspective, but give only marginal insight into how new data sources can be interfaced to AIAT.

The AIAT documentation stresses thorough design and analysis. Having a clear picture of which objects deal with which data and where that data goes will make your coding efforts much more enjoyable. Now that you've done that, let's get down to architecture. AIAT functionality is broken up into six major categories: *Index*, *Analysis*, *Corpus*, *Accessor*, *Storage*, and *Storable*. The Index classes handle creation and maintenance of keyword lists and document references. The Analysis classes are responsible for generating and filtering keywords based on

When he isn't messing about in MacsBug, **Mark Holtz** is one of the founders of MacISP, helping people and organizations provide Internet services using the Mac OS. E-Mail to mholtz@intermag.com should garner a quick response, unless he's out having coffee.

various criteria. The Corpus classes are an abstraction layer for obtaining data from various sources. The Accessor classes provide query and statistical information from indices. Finally, the Storage and Storable classes provide a flexible mechanism for the storage and retrieval of arbitrary sets of data. A quick browse through the headers shows an abstract C++ object for each of these categories, such as the class `IACorpus`.

You will also find various utility functions, including a set of memory management calls. You can substitute your own allocator and deallocator routines easily, and AIAT will happily use them for block and object allocations. There are a few robust concrete subclasses provided, which you can use directly or subclass to enhance their functionality. The `HFSCorpus` and `HFSTextFolderCorpus` classes allow AIAT to index collections of text documents. The `EnglishAnalysis` class allows AIAT to filter document keywords with arbitrary stop-word and word-stem lists. AIAT makes use of its own exception handling mechanism, based on C++ exceptions with a few additions. A little exploration through all the components of the AIAT distribution is time well spent, as there are a number of classes and functions not specifically called out in the documentation.

THE TASK AT HAND

I was first introduced to AIAT early in 1997 during a conversation with a good friend of mine. He suggested that I look into it as a possible full-text indexing solution for a Mac-based Web site we were creating. As the months passed, we settled on Purity Software's WebSiphon product as the back-end scripting engine for this Web site. WebSiphon includes a fast flat-file database called Verona. It had basic search capabilities, but they were not robust enough to support the kind of ranked queries you'd find on more powerful Full-Text Indexing Systems. WebSiphon's language is extensible via Code Fragment libraries, so the idea for an AIAT library for WebSiphon became an appealing solution. We could collect information from a variety of sources with WebSiphon, store it in Verona, index it using AIAT, and then query against that data using WebSiphon scripts. Since AIAT is simply a set of libraries with little runtime environment dependency, the task was not daunting. The project was distilled into a few discrete tasks: Create the scripting interface for WebSiphon, write the C++ code to interface with the AIAT accessor and index classes, write the C++ code for interfacing AIAT to Verona, and make the entire thing work in a multi-threaded environment. Each phase of the project involved one of AIAT's major areas of functionality, so I was able to focus on one set of concepts at a time, a tribute to AIAT's modular design.

Habeas Corpus

The first task to tackle was to provide AIAT with an interface to Verona so it could access the realms of data our site would produce. This involved getting familiar with AIAT's Corpus classes. In AIAT, the Corpus provides access to a collection of "documents" and the data they contain. The `IADoc` class is the abstract representation for these documents, and it consists of a name for the document and access methods for the data. In this

dtf SQL

A relational database for product developers!

Need true relational database functionality that doesn't cost an arm and a leg? Introducing **dtf/SQL**. Fast, efficient, proven, truly relational. Why bother using ISAM databases now that SQL technology finally has become affordable? Or even free for non-commercial use? Don't wait, stop shuffling records, and let **dtf/SQL** handle your database worries.

Powerful features

- ◆ Full transaction handling ◆ automatic locks
- ◆ referential integrity ◆ advanced R4 recovery with full transaction logging ◆ passwords, access rights, encrypted databases ◆ standalone, client, and server versions available ◆ query optimization and load balancing ◆ small memory footprint
- ◆ C/C++ interface, ODBC driver, EOF-Adapter, Hypercard, Supercard, and other interfaces

Cross-platform & cross-protocol

- ◆ Mac OS ◆ Rhapsody ◆ DOS ◆ Windows 3.x
- ◆ Windows 95/NT ◆ OS/2 ◆ Novell NLM
- ◆ several UNIX flavors, including Linux ◆ TCP/IP
- ◆ IPX ◆ NETBIOS ◆ AppleTalk ◆ Named Pipes

Great licensing terms

- ◆ Use for free during development
- ◆ free shipping license for non-commercial use ◆ low rates for commercial products
- ◆ Special agreements for CD-ROM apps available!

New
dtf/SQL 1.7
also for
Rhapsody



Go check our WEB site.
<http://www.slub.de>

case, our documents were records in the Verona database, and the data would be accessed via an API to the Verona application instead of reading it directly from a file. After a little bit of work with the Code Fragment Manager, I had a functional C++ interface to Verona, so it was time to tell AIAT how to deal with it. First, I created the CVeronaCorpus class, and defined the two pure virtual methods required to make it work: GetProtoDoc and GetDocText.

Listing 1: CVeronaCorpus.h

CVeronaCorpus

Class definition and required methods for our Corpus interface to a Verona database.

```
class CVeronaCorpus : public IACorpus
{
public:
    CVeronaCorpus(CVeronaGlue* inVeronaGlue,
                  const char* inDBName);
    virtual ~CVeronaCorpus();

    virtual IADoc*      GetProtoDoc();
    virtual IADocText*  GetDocText(const IADoc* doc);
    virtual IADocIterator* GetDocIterator();

    .
    .
    .
};

IADoc* CVeronaCorpus::GetProtoDoc()
{
    return new CVeronaDoc(this, 0);
}

IADocText* CVeronaCorpus::GetDocText(const IADoc* doc)
{
    return new CVeronaDocText(this, (CVeronaDoc*) doc);
}

IADocIterator* CVeronaCorpus::GetDocIterator()
{
    return new CVeronaDocIterator(this);
}
```

Phew! That wasn't so bad. GetProtoDoc() returns a new CVeronaDoc object, and GetDocText() returns a new CVeronaDocText object. But what are these objects? CVeronaDoc is a subclass of IADoc, AIAT's abstract representation of a document within a Corpus. CVeronaDocText is a subclass of IADocText, which is responsible for providing the actual text of the document to AIAT's indexing functions. The third method, which is not required by AIAT to make a valid Corpus mechanism, is GetDocIterator(). The IADocIterator class is used to implement a Corpus that consists of multiple documents, and for providing each of those documents to AIAT in a consistent, ordered fashion. You may notice as you peruse the AIAT documentation that many functions deal with IADoc's as a fundamental unit of data. It is up to the Corpus to determine what that unit of data is and how to return it to AIAT when it's requested. Here are the definitions of the other Corpus subclasses I created:

Listing 2: CVeronaCorpus.h (cont'd.)

CVeronaDoc, CVeronaDocText, CVeronaDocIterator

Classes for representing the various sub-elements of CVeronaCorpus.

```
class CVeronaDocIterator : public IADocIterator
{
public:
    CVeronaDocIterator(CVeronaCorpus* inCorpus);
    virtual ~CVeronaDocIterator();
}
```

```
virtual IADoc* GetNextDoc();

private:
    CVeronaCorpus* mCorpus;
    unsigned long mCurrentIndex;

};

class CVeronaDoc : public IADoc
{
public:
    CVeronaDoc(CVeronaCorpus* inCorpus,
               unsigned long inRecRef);
    virtual ~CVeronaDoc();

    IASortable* DeepCopy() const;
    IABlockSizeStoreSize() const;
    void Store(IAOutputBlock* output) const;
    IASortable* Restore(IAInputBlock* input) const;

    bool LessThan(const IAOrderedStorable* neighbor) const;
    bool Equal(const IAOrderedStorable* neighbor) const;

    virtual byte* GetName(uint32 *length) const;
    unsigned long GetRecRef(void) { return mRecRef; }

protected:
    virtual void DeepCopying(const IASortable* source);
    virtual void Restoring(IAInputBlock* input,
                          const IASortable* proto);

    .
    .
    .
private:
    CVeronaCorpus* mCorpus;
    unsigned long mRecRef;
};

class CVeronaDocText : public IADocText
{
public:
    CVeronaDocText(CVeronaCorpus* inCorpus,
                   CVeronaDoc* inDoc);
    virtual ~CVeronaDocText();

    virtual uint32 GetNextBuffer(byte* buffer,
                                uint32 bufferLen);
    virtual IADocText* DeepCopy() const;

protected:
private:
    CVeronaCorpus* mCorpus;
    CVeronaDoc* mDoc;
    byte* mBuffer;
    unsigned long mAmtRead;
    unsigned long mBufSize;
};
```

At this point, we've got all the elements for our Corpus implementation. However, it may still be unclear how these items work together. When AIAT receives a request to update a particular index, it starts a dialog with the Corpus object that is tied to that index. It starts by asking, "What sort of documents do you contain?" By calling GetProtoDoc(), the Corpus can supply AIAT with a "sample" document. AIAT then asks for an object that can iterate through all the documents in the Corpus' collection. If one is available, it is returned via the GetDocIterator() method. Since AIAT knows nothing about the particular data set it's indexing, the Corpus needs to provide these mechanisms. If a document iterator is available (which is true in this case), AIAT begins asking the iterator for successive documents in the collection.

AIAT makes two assumptions about documents that you must keep in mind. First, all documents in a particular Corpus are of the

same type (i.e. `CVeronaDoc`), and second, that the order of the documents is always the same for a particular set of documents. The latter is important because AIAT uses the document sequence for the indexing mechanism. Hence, the notion of a document being "Less than" another document really has to do with its order in this sequence. Be sure to be consistent for whatever kind of data you're delivering, and this should not be a problem. Getting back to the dialog, AIAT now has an `IADoc` it can work with. It starts by asking the document for an `IADocText` object that contains the document's data. In our example, the `CVeronaDocText` class knows how to access the text in a Verona database record, so our `CVeronaDoc` object hands a fresh `CVeronaDocText` object back to AIAT. This object can access its parent `CVeronaDoc` object, and uses that link to obtain the record number in the database that the `CVeronaDoc` represents.

Finally, it is time for AIAT to get the text of the document. It does this by calling `CVeronaDocText`'s `GetNextBuffer()` method. This method returns the specified number of bytes from the document. Note that the `CVeronaDocText` object must maintain its own information about what data AIAT has already requested. It may help to think of `CVeronaDocText` as a one-way stream of data that is read in chunks of arbitrary size. AIAT will continue to call `GetNextBuffer()` until the method returns zero, indicating the end of the data.

AIAT will continue to call the iterator and resulting document and document text objects to obtain the complete set of data in the Corpus' collection. The other methods of `IADoc` are used to determine how a particular document should be placed in the index. It is necessary to override these in your subclasses so that they are meaningful to the data you're representing. In my case, the `LessThan()` and `EqualTo()` methods compare documents based on their specific Verona index number. Also amongst the methods of `IADoc` you'll have to override are `Store()`, `Restore()` and `DeepCopy()`. These methods handle converting the object into a data stream, creating an object based on a data stream, and making a complete and independent copy of an object. These functions are nicely explained in the AIAT documentation, except for the `DeepCopying()` and `Restoring()` methods, which are used to construct the superclasses of your document class in a `DeepCopy` or `Restore` situation. The only place these functions seem to be documented is within the header file of `IADoc` itself. This may be a perfect time to go back and browse the AIAT headers again.

Now we have a complete Corpus structure for AIAT to handle Verona Databases. A bit of implementation here and there, and we're ready to start indexing and querying the data. However, before moving on to the next section, it is important to keep in mind that there is a lot of possible functionality within the Corpus classes that I have not covered here. For instance, one can filter documents selectively within their `IADocIterator` subclass, providing only the documents they want to AIAT. As an example, the `HFSTextFolderCorpus` class filters out files that aren't of type "TEXT". For the Verona interface, it was not necessary to implement a very complex Corpus interface, but be sure to explore the section fully and determine which features can help you in your application.

MkLinux

Microkernel Linux for the Power Macintosh

MkLinux is a complete system, based on Linux 2 and the Mach 3 microkernel. It includes development tools (gcc, gdb, perl, ...), X11R6.3, and hundreds of other commands.

MkLinux builds and runs most Intel-based Linux software. It works efficiently and reliably on a wide range of Power Macintosh platforms. Visit Apple Computer's MkLinux web site (www.mklinux.apple.com) and Prime Time Freeware's web site (www.ptf.com) to find out more about MkLinux!

MacPerl: Power and Ease

MacPerl is a robust and powerful port of Perl 5 to the Mac. It runs under the Finder and MPW, supports Apple Events and Toolbox calls, and is generally quite nifty! For details on MacPerl and associated products (book, CD, etc.), visit the MacPerl Pages (www.ptf.com/macperl).

Prime Time Freeware
370 Altair Way, #150
Sunnyvale, CA 94086

info@ptf.com
(408) 433-9662
(408) 433-0727 fax

Indexing and You

The ultimate goal of AIAT is to generate an index of terms that can be used to satisfy queries about the database. AIAT provides two different indexing mechanisms to accomplish this goal. An Inverted Index contains a list of terms and references to documents that contain the term. A Vector Index contains a list of documents and the terms that each document contains. These two types of indices can be used separately or together to satisfy various types of queries. In AIAT, the two types of indices are combined to create a Ranked Index, capable of providing ranked results to keyword, boolean, and example-document queries. For the purposes of this project, we chose to use a Ranked index because we would be searching our data in a variety of ways, and wanted to provide as many possible methods as we could.

The next task in our development project was to actually index the data using AIAT. This is accomplished by using the Index, Corpus, Storage, and Analysis classes in concert. The code is simple and straightforward, and the examples are an excellent place to start. Here's a snippet of our indexing code, which is a slightly modified version of the example code:

```
IATry
{
    tIndex = mDataSource->GetIndex();
    //
    // Makes calls to:
    // MakeHFSSStorage(), new CVeronaCorpus(dbName),
    // new SimpleAnalysis(), and new InVecIndex()
```



```

tStorage = mDataSource->GetStorage();

tStorage->Initialize();
tIndex->Initialize();

tIndex->SetFlushProgressFn(
    &CSTwinDSUpdateThread::sFlushUpdateFunc);
tIndex->SetFlushProgressFreq(10);

tIndex->Update();
tStorage->Commit();
}
IACatch (const IAEException& exception)
{
    ::SysBeep(0);
    WSAppendLog((char*) exception.What());
}

```

To create an index, a few other objects need to be created. First, we need to store the Index data somewhere. This is accomplished with the `IStorage` class and its subclass, `HFSSStorage`. AIAT provides a built-in mechanism for creating disk-based `IStorage` objects, so one call to `MakeHFSSStorage()` takes care of this task. Next, we need to establish a Corpus to index. Creating a new `CVeronaCorpus` object that points to the desired Verona database satisfies this requirement. The third part of the puzzle involves the method AIAT should use to analyze the Corpus' data. The `IAAnalysis` class is used for this operation, and the provided `SimpleAnalysis()` class was fine for our immediate needs. It eliminates words under three characters in length, and changes all terms it finds to lower case letters. Finally, we create the Index object itself. Since we are using the Inverted Vector index, a call to `new InVecIndex()` with the pointers to the storage, corpus, and analysis objects will suffice nicely. In this project, I've wrapped most of the specific creation into the `GetIndex()` method, and it takes care of instantiating anything it may need.

Initializing the index is our next task. Since we're creating this index from scratch, we call the `Initialize()` method of our `IAIndex` object. If we wanted to update an existing index, we would call the `Open()` method instead. AIAT does support incremental updates of an index, but it requires you to write some more logic in your Corpus classes. Due to limitations in the Verona database structure, we couldn't provide a clean method for determining when records had changed, so we didn't write the extra code to support incremental updating. To properly initialize a new index, you must first initialize the Storage, and then initialize the Index, using their respective `Initialize()` methods. The index is now cleared and open, awaiting new terms and data from the Corpus.

At this point, I will make quick mention of the various progress functions in AIAT. They are available at most places in the code where procedures may take some time, such as during indexing and long queries. These functions are well documented, and behave exactly as you would expect. Following the progress function setup calls, we find a call to the `Update()` method of our index object. This sets the entire AIAT indexing mechanism into motion. AIAT conducts the dialog with the Corpus that I outlined earlier, then takes the resulting data, processes it with the `IAAnalysis` object we provided, and stores the information in our Storage object. Once the process is complete, the data is flushed to the storage object via `IStorage::Commit()`.

One last thing to note in this code snippet is the use of the `IACatch` and `IATry` macros. These expand to more standard C++ exceptions (or to other exception handling devices if C++ exceptions aren't available). AIAT uses the Try, Catch, Throw model for most all error reporting. If anything goes wrong, such as an index couldn't be opened or you run out of disk space, the program's execution would jump immediately to the `IACatch` block of code. In this example, we just beep and make a note in the log, but you will probably want to be more robust in your handling of errors. If you are unfamiliar with exception handling in general, I would strongly recommend reading up on it before you start your AIAT-related project.

Live to Search, Search to Live

Whenever I develop a new piece of software, I like to set my sights on an end result that I can work towards, and when it happens, I'll know I've made it. In this case, the end result was to see a Web page that showed database records ranked in relevance to my query string. We've made it through interfacing a new data source to AIAT, and asking AIAT to index it for us. Now we need to make that index produce something useful. AIAT refers to a document that match a query as a "hit".

To generate a list of hits in response to a query, we bring a new class, `IAAccessor`, into play alongside `IAIndex` and `IStorage`. The Accessor class ties one or more indexes together and provides methods for posting a query and generating lists of hits. It also contains several small classes for describing hits, including the `RankedHit` class which contains a document reference and a percentile ranking, and the `RankedQueryDoc` class which can be used to formulate Query-by-Example functions in AIAT. One of the more interesting features of the Accessor class is that it allows you to search several indices at the same time, and the results can be a mix of documents from several different types of data sources. In our project, we support Verona databases, FileMaker databases, folders of text documents, and a few other types of data sources. With AIAT, we can search any combination of data source types, and provide document references for each hit, along with specific information about the data source that contains the particular document.

The following snippet is from the code to generate a ranked hit list in response to a basic keyword query.

Listing 3: Perform a ranked query on multiple data sources

rankedQuery_fn

```

char* tQuery = "apple internet"; // query string
unsigned long count = 25; // max # of hits
unsigned long resultCount = 0;
unsigned long i, tCt;
HFSDoc tDoc;

RankedHit** results = IAMallocArray(RankedHit*, count);

InVecIndex** indices = IAMallocArray(InVecIndex*, tCt);

// Setup indices[] with list of indexes to search.

```



```

IATry
{
    for (i=0;i<tCt;i++)
    {
        indices[i]->Open(true);
    }

    CSTwinAccessor accessor(indices, tCt);
    // MacroInitialize takes care of various init states properly
    accessor.MacroInitialize();

    resultCount = accessor.RankedSearch(
        (unsigned char*) tQuery,
        strlen(tQuery),
        NULL, 0,
        results, count, 4,
        NULL, 30, NULL);
    DisplayResults(sources, tCt,
        results, resultCount);
}
IACatch (const IAEException& exception)
{
    char tErr[255];
    sprintf(tErr, "Caught exception: %s\n",
        exception.What());
    WSAppendLog(tErr);
}

for (i=0;i<tCt;i++)
    sources[i]->Close();

for (i=0;i<resultCount;i++)
{
    delete results[i];
}

IAFreeArray(results);
IAFreeArray(sources);
IAFreeArray(indices);
return noErr;

```

PrimeBase™

Prime Power Online

High-performance Relational DBMS

Now available for Rhapsody

Open System: ODBC, SQL, DAL Compliant

Scalable from 68K Mac to IBM RS6000

Proven Reliability (formerly P.INK SQL) with installations around the world

Cross-platform: Rhapsody, MacOS, Windows 95/NT, Sun Solaris, Linux, ...

Competitive Prices (10-user server for Apple Hardware or PC only \$820)

Highly Flexible **Web Connectivity** available

EOF Adapter for OpenStep and Rhapsody

Single-user **Runtime** available

Developer Keys **Free!**

SNAP Innovation GmbH
Tel : +49 (40) 30636400
Fax: +49 (40) 30636333
sales@primebase.com

Download Now!
www.primbase.com



Macintosh Programmers and Developers... PRODUCE and DUPLICATE quality disks Quickly and Economically with the "CLONE-IT"



"CLONE-IT" 1 to 3
\$2,195

STARTING UNDER \$1,000 "The XEROX Machine for CDs"

- 3 in 1 solution of having a CD-Duplicator, a Philips 2x/6x Writer (3 in the 1 to 3 Duplicator model), and a 12x CD-ROM in one unit.
- Stand-Alone capability for performing its function as a CD-Duplicator. Produces absolute identical bit-by-bit copies.
- Easy hook-up to MAC systems. Automatic CD format detection and transfer to CD-R media.
- Simple operation with the touch of a button.



"CLONE-IT" 1 to 1
\$995

CD-Recordables

with "Printable Surface"
on Spindles of 100 Blank CDs



\$1.85

Optics Storage Pte. Ltd. - USA

1162 St. George Ave., Suite 150, Avenel, N.J. 07001-1263

Tel: (732) 636-4466

Fax: (732) 750-1793

E-mail: optixsusa@aol.com

Visit Our Web Site For Details On How To Receive A **FREE** CD Duplicator - **www.optixs.com**

In this snippet, we find that the logic required to generate hits with AIAT is very straightforward. We create arrays for our results and our source indices using `IAMallocArray()`, which is just a macro for AIAT's allocator. We can use our own memory allocator if we want to, and can have AIAT use it by setting one of AIAT's public variables to the desired function. This is outlined in the documentation and the example code. It is important that you allocate and deallocate AIAT classes and structures with the same allocation code, or you may find yourself in trouble. However, AIAT subclasses all of its objects from one root class, `IAObject`, which has provisions within it to always allocate and deallocate using AIAT's allocator functions. Since you can set them to whatever you want, AIAT will behave itself in whatever memory allocation environment you may have.

Next, we find ourselves inside of another `IATry` block. This is because any AIAT calls from this point on may generate an exception, and we need to handle them appropriately. We run through the list of indices we want to search, and open them. The code for gathering the indices was removed for clarity, and it's up to you as to when and how you create the index objects for use in a search.

The next few lines were the most problematic in this project. Although it looked simple in the example application, I had overlooked an important piece of information when I read the documentation. The `IAAccessor` class requires a certain amount of information to be properly initialized. The example application tested for the presence of this information in the index's storage, and if it wasn't there, it would create it. If it was there, it would simply load it up and the accessor would be ready to use. I missed this line of code, and subsequently, all my searches would crash my machine in strange and wonderful ways. So, I created my own subclass of `InVecAccessor` that simply had this additional function to properly initialize an accessor in any situation.

Listing 4: Proper initialization of an accessor regardless of previous state

CSTwinAccessor.cp, CSTwinAccessor::MacroInitialize()

```
void CSTwinAccessor::MacroInitialize()
{
    IAIndex** tIndices = ::GetIndices();
    unsigned long i;

    if (!IsInitializationValid())
    {
        StoreInitialization();

        for (i=0; i<GetIndexCount(); i++)
        {
            IAStorage* tStorage = tIndices[i]->GetStorage();
            tStorage->Commit();
        }
    }
    else
        Initialize();
}
```

This function uses `IAAccessor::IsInitializationValid()` and `IAAccessor::StoreInitialization()` to figure out the current state of the Accessor object and store it each of the indices' Storage. Don't forget to call `Commit()` for each storage object to flush the changes.

Following the successful initialization of the accessor, we are ready to make our query. We ask the accessor object to return hits that match `tQuery` in the array of `RankedHit` object we created earlier. `ResultCount` is set to the number of actual results in the array, and I then call the `DisplayResults` function to display the resulting hits. The AIAT example code shows how to access the various items within a `RankedHit` object. Using the `Index` and `Doc` components of a `RankedHit`, you can figure out which index a hit comes from, and thus what type of document it is. You can then cast the `IADoc*` to an appropriate `IADoc` subclass, and access additional information when presenting your results.

Finally, we close up the indices and delete the `RankedHit` objects that the accessor created for us. With a few `IAFreeArray()` calls to go, we have completed our first query against our custom data set using AIAT. That didn't hurt so much, did it?

WHAT TO DO FOR AN ENCORE

It would be hard to encompass all the facets of AIAT within the scope of a single article. With just a small amount of implementation, however, you can interface AIAT to just about anything. As with any technology, the best thing you can do is explore and try different things. (Isn't that what the Macintosh is all about?) Once you're comfortable with the basic concepts in AIAT, there are several concrete classes for you to try out, such as `English Analysis` (or `Korean Analysis`!). The `IAAnalysis` class and its subclasses contain a great deal of interesting functionality, and since they control which terms get generated by AIAT, they are extremely important for ensuring that once you've got your search engine running, the data it returns is actually useful.

To echo the words of the AIAT manual, be sure to spend a good amount of time understanding both the AIAT architecture and the problem you're actually trying to solve. Whatever may be lacking in the documentation about implementation details is more than made up for by the walk-through examples that are provided. The AIAT architecture is fairly orthogonal, so once you understand the behavior of one class of objects, you will find that the other classes follow suit. For the power it provides, AIAT provides a lot of benefit without a great deal of implementation effort.

RESOURCES

- 1) Apple Developer CD Series, Tool Chest/Mac OS 8 Edition, November 1997.
- 2) <<http://www.research.apple.com/research/tech/V-Twin/>>.
- 3) Apple Developer World, <<http://www.devworld.apple.com/>>.
- 4) Purity Software, Inc. at <<http://www.purity.com/>>.

17

Interested in writing for the magazine?
Contact us about a writer's kit at
 <<mailto:editorial@mactech.com>>

by William A. Gilbert, Ph.D.

dtF/SQL — The Little Engine That Could

*sLab makes there powerful
SQL database engine
available for anyone to
use, at little or no cost*

INTRODUCTION

File formats, saving data, relating fields, multiple files versus single file, backward compatibility... this is the struggle of all programmers with each new project. One could always use a third party database product such as FileMaker Pro or 4D but that only increases the complexity of development, deployment, and support. Another alternative is to use a centralized database system such as Sybase, Oracle, or Microsoft SQL Server, but often these solutions go beyond the scope of the project and introduce even more complexity and support problems.

Often what is needed is a small library which can store data, allow for the addition of new data definitions while being able to provide reasonable backward compatibility and to allow data protection from your competition. Yet at the same time it must allow access by those very special people — the clients and customers!

Enter dtF/SQL, the little relational database engine that not only could but does achieve these goals and many others. dtF/SQL, a relational database engine for Mac OS, Windows 95/NT, and several UNIX

platforms from sLAB <<http://www.slab.com/>>, is provided as a small (403 Kbytes) linkable library which provides pure performance and pure pleasure for any developer who requires data storage in a file.

Having worked extensively with large scale SQL databases I had always wanted a product such as dtF/SQL for the "little" projects. With this library linked directly into my application, I can store both data and program parameters, access that data using simple or complex queries, and extend the data format by either adding additional fields to an existing table or by creating new tables. Using Finder file type and creator signature tags I can create double-clickable documents which launch the appropriate application under the Finder, as well as create custom icons for these documents. And the best part... it's free to non-commercial users and has reasonable licensing for commercial use. This is explained comprehensively at their web site.

dtF/SQL implements an impressive set of ANSI SQL functionality. It also supports the BLOB data type which is an arbitrarily large binary stream of data such as an image or a numeric array of data values. Both local and remote retrieval is fast. This subjective statement is based on experience with other database engines as well as experience with reading and parsing other file formats.

HIDING COMPLEXITY

Our goal as programmers is to provide our clients and customers with a product which performs both to specifications and expectations, but is also easy to use. Grady Booch has put it well: "The task of the software development team is to engineer the illusion of simplicity." This statement is particularly true for Macintosh applications, where end users have come to expect an environment in which one copies the application with some documents from the distribution disk and just expects to double-click a document and go. Using the dtF/SQL engine there are no extensions to load and configure, there is no restarting the computer, and there is no performance overhead imposed by switching processes or drivers.

William Gilbert <<http://www.informagen.com/cv/Gilbert.html>> is a biochemist by training as well as a computer professional. He provides informatics solutions and scientific consulting to startup biotechnology companies. He also holds a research professorship at the University of New Hampshire in the Department of Biochemistry and Molecular Biology.

Additionally, there is no middleware to install with additional royalties. The bottom line is that the developer is able to deploy self-contained double-clickable database applications and documents, runnable even directly from a CD-ROM.

ONE PROGRAMMER'S EXPERIENCE

I found the dtF/SQL API definitions and documentation, provided in PDF format, to be very readable. There are over 400 pages which describe the installation, data types, high level API, and low level API. Each API has a usage example as well as any restriction or limitation of the API.

The Macintosh distribution comes with the application, dtF/Admin, which allows one to create and experiment with a database to see the effects of SQL queries. This tool is a great way for beginners to learn how to create database schema or test SQL queries on real data before committing these queries to source code. If you mess up the database, just throw it in the trash and start over; you won't have an enraged database administrator breathing down your cubicle. The dtF/Admin application allows you to save scripts so that the database can be recreated and repopulated in seconds.

Of course, this can be done under program control as well. I keep my database schema scripts as STR# resources, then, with a New command, a new database can be created anytime.

At first I was disappointed that I couldn't write the equivalent of Sybase stored procedures to facilitate queries. I then realized that I was equally well off not being able to write and invoke stored procedures, because I was able to implement these procedures as C or C++ routines. I found that where I had

become accustomed to writing complex SQL **select** statements with extensive join clauses, I could, with dtF/SQL, create a couple of separate workspaces, then use nested **for** loops to perform the same queries in a faster, much more logical fashion.

For those of you not familiar with SQL, the implications of this are significant. The SQL join command to resolve relationships takes lots of memory. In contrast, successive **select** statements which use the results of one SQL statement as input to the next SQL **select** statement takes almost no memory at all and with dtF/SQL is probably faster than doing the original join. The bottom line here is that one does not have to become an SQL guru to use dtF/SQL effectively. Arguably, such expertise may even work against you. The web-based, downloadable distribution comes with several complete, non-trivial example programs which demonstrate the usage of almost all of the APIs.

IF ONLY...

dtF/SQL's implementation departs from the Macintosh application & document paradigm in one respect. In release 1.6 the database is actually two files, a database file and a separate file used to store BLOB data. This can present some challenges to the double-clickable document interface. The application must determine the location of the other file and then open it. For situations where a BLOB file is not required, an empty BLOB file can be created on the fly and opened, then deleted when the database session is over. If a BLOB file is required, one can use a Windows and UNIX trick and require that the data file and BLOB file have the same name but require that the BLOB file end with the word "BLOB". Not truly the Macintosh way of doing things, but it works well enough.

The requirement of a separate BLOB file will be removed in release 1.8 which is expected to be available in early 1998. Then BLOB data will be stored with all other data in the single file.

AVAILABILITY

The dtF/SQL object library is available for Macintosh System 7.x and Mac OS 8 (68K and native PowerPC). MPW C/C++ , Symantec C/C++ and CodeWarrior C/C++ are supported. Separate versions for HyperCard, SuperCard, AppleScript, MacHTTP CGI, Smalltalk Agents and other environments are available. dtF/SQL also is an integral component of the Internet-enabled client/server development system from Pictorius, Inc. and the Oberon/F-based client/server system from microsystems. The dtF/SQL engine is also available for DOS, Win16, Win32, NT, OS/2, SUN OS, HP/UX and Linux. You must purchase support separately. It is provided by e-mail and is timely and informative.

The documentation is supplied as PDF files which can be read with Adobe Acrobat. I would have preferred hard copy, especially for the API definitions when first learning the library. On the other hand, these APIs are few and are learned quickly. The code examples are excellent and non-trivial. There also are some third party examples which range from obvious to obfuscated.

The CodeWarrior CD-ROM distribution contains a demo version of this product, dtF Lite, as well as some experimental PowerPlant classes which encapsulate the dtF/SQL API. The

**Want to share a tip with
the community and
get paid for it?**

**Send it in to
<mailto:tips@mactech.com>**

standalone database tool, dtF/Admin, also is provided and is very useful of anyone who wants to learn more about relational databases and the SQL syntax in general

The database and BLOB files are binary compatible between Macintosh, PC, and UNIX platforms, making the dtF/SQL engine and database files a perfect candidate for distribution on CD-ROM.

There are ODBC drivers for dtF/SQL for Win16 and Win32; Macintosh ODBC drivers will be ported shortly. Support for Java JDBC is under development and should be part of the upcoming version 1.7 release.

SCALABILITY

Once you have deployed your application, your users may come back and ask about scaling up so that the members of their organizations can use your application to examine and modify shared data. Does this mean that you will have scrap your work and learn to use Sybase, Oracle, or ODBC libraries? No! dtF/SQL is available in a client/server configuration; so, as a developer, you add a few lines of connection code and relink your existing source with the dtF client library. Your customer purchases and runs the dtF/SQL server for their database files (licensed separately from sLAB, downloaded from the web with a password). They can now use your newly relinked application, with its familiar UI, to access and share their data. The dtF/SQL LAN server runs on Macintosh, Windows 95/NT, and some UNIX platforms. Version 1.7 will allow any dtF/SQL client to connect to any dtF/SQL server.

RECOMMENDATION

I can unconditionally recommend the dtF library to any programmer who wants to develop and deploy solutions which deliver performance and simplicity to the customer. As for robustness, let me add in conclusion that I have crashed about a thousand times during development and have never had a dtF/SQL database file become corrupt as a result.

PRICING

All single user non-commercial dtF/SQL products are free and can be downloaded from the web, including documentation. There are no restrictions such as time-outs or maximum record sizes with the downloaded product. Non-commercial uses of the engine are also free. Distribution of commercial products are priced on a unit basis. Large volume pricing can bring the price down to 1 DM (about \$0.65 US) per unit. For server pricing contact sLAB directly by e-mail or phone, available from their web site.



USEFUL URL'S

<<http://www.slab.de/uk/homepage/index.html>> is sLab's home page, in English. All dtF products, commercial and non-commercial, are available from this site.

StoneTable

You thought it was **just** a replacement
for the List Manager ?

We lied, it is **much** more !

Tired of always adding just one more feature to your LDEF or
table code ? What do you need in your table ?

Pictures and Icons and Checkboxes ?
adjustable columns or rows ?
Titles for columns or rows ?
In-line editing of cell text ?
More than 32K of data ?
Color and styles ?
Sorting ?
More ??

How much longer does the list need to be to make it worth
\$200 of your time ?

See just how long the list is for StoneTable.

Make StoneTable part of your toolbox today !

Only \$200.00

MasterCard & Visa accepted.

StoneTable Publishing

More Info & demo

Voice/FAX (503) 287-3424

<http://www.teleport.com/~stack>

stack@teleport.com

Editor's Note

sLab has made an astonishing offer to the programming community at large — a free, high end SQL relational database for non-commercial use. After I read this review, I trotted over to their home page and downloaded all the stuff I needed to look at this product. As Dr. Gilbert indicates, this is a super package. This is a great opportunity to learn and have fun with no investment, and presumably sLab will reap the appropriate benefits when we all write our killer apps based on their libraries.

I had never dabbled around with an SQL-based package before, and asked Will to recommend an introduction to the language. The book he recommended is quite readable and informative. It's only shortcoming is that the accompanying CD is Windows only, but many of us now have a Wintel machine or an emulator program, and I won't tell anyone if you won't. :-)

The Practical SQL Handbook, third ed., 1996, by Bowman, Emerson, and Darnovsky. Addison Wesley Developers Press, Reading MA. ISBN 0-201-44787-8.

—Ed Ringel

by Edward Ringel

Jovis, a Database Engine in a Tiny Package

A robust database engine for HyperCard and other XCMD compatible environments

INTRODUCTION

Knowledge base development, particularly contract or in-house development, does not require a C, Pascal or C++ environment. To the contrary, many corporate databases and commercially deployed knowledge bases are well supported by database languages such as Fourth Dimension. Other knowledge bases use an authoring front end along with an attached database engine. For example, I use a product called Scientific American Medicine, a complete electronic textbook of internal medicine, that is extensively cross referenced and searchable. It was developed with a MacroMedia Director front end. High level, programmable authoring environments such as HyperCard, SuperCard, and Director can be particularly appropriate when the knowledge base includes pictures, sounds, and videos. These environments are inherently multimedia capable and require little programming knowledge to play a QuickTime clip or show a PICT.

Jovis, by DAS Works, is an extended XFCN that provides broad relational and

architectural database capabilities for environments that support XCMD's and XFCN's. This is a comprehensive, well documented product that can support knowledge base development and more traditional database functions in a number of different user settings.

OUT OF THE BOX

I received a single disk, a serial number, and a big book. The product comes as a single user SDK or an individually configured Client/Server product; I was sent the single user product for review. In order to use Jovis, I needed to install the product into a HyperCard stack. I could not install the XFCN's into HyperCard itself. Installation, rather than execution is protected by the serial number. (Licensing is addressed later.) The code I used is 68K based and is just under 400K. A PPC version is also available if requested.

The documentation consists of a 16 lesson tutorial and extensive documentation of the various calls. As with most good documentation, the tutorial builds nicely upon itself and is easy to follow. DAS Works does not supply step by step "solutions" to the tutorials; this forces the reader to actually work through the projects to see if they work. The demo supplied with the product actually is a culmination of the tutorials and is a real world example. The lessons address both the relational and architectural capabilities of the product. An introduction also is provided to show some of the intricacies of multi-user issues such as record locking and transactions.

The reference section comprehensively addresses the Jovis calls. Each entry describes the syntax of the call, provides an example, describes the action of the call, and offers relevant comments.

Although a tutorial for Jovis is provided, there is little in the way of more complex examples or a section on "putting it all together." I think they must assume that the user has a good idea of how to use a database. The tutorial teaches the nomenclature

Ed Ringel is Contributing Editor for product reviews for *MacTech Magazine*. When he's not working at the computer or enjoying the Maine lakes and woods, he's a respiratory and critical care physician in Waterville, Maine. He can be reached at erinkel@mint.net

and specifics of this particular environment, but does not address design or the management of a complex environment.

THE PRODUCT

Jovis supports a comprehensive set of relational commands and structures. Relational file creation and open/close operations are very simple and straightforward. Physical files on disk are referred to as "collections," and can contain multiple tables ("relations") and indices. Architectural files are flat file structures that allow the storage, indexing, and rapid retrieval of arbitrary size binary objects (Binary Large Objects, or BLOBs.)

Relation and index construction within a collection requires some work and programming, as with any database system. Jovis offers a shortcut to field and index creation by permitting the creation of a field list or index list. Conceptually, this is like a STR# resource that is then read into the create function, but it is supported as a text string within the scripting language of the front end environment.

Reading and writing records can be performed from script variables or directly into fields in a card or background. Large amounts of data can be read directly into records by an ImportData command. This can be particularly valuable for filling a knowledge base once the structure has been designed and created. There also is an ExportData command.

Obviously, an important feature of a relational engine is the ability to perform a complex search. Jovis does this well. There is a nice section in the tutorial that describes the various comparison operators and how to create compound selection criteria. Jovis supports range searches and the creation of selections. Multiple selections for a given relation are permitted. Merges, which are the same as SQL joins, are supported as a two step process. Selections are first created and then merged. The tutorial completely explains this process.

Record locking and transactions are supported for the multiuser version. There is a section on how the Jovis server works at a file level, but the manual does not teach you how to undertake the design and deployment of a shared, multiuser database system. Calls provided, however, will permit the knowledgeable user to implement a robust, safe multiuser database.

The architectural commands are as important as the relational instructions. Multimedia developers will make heavy use of BLOBs and managing these objects is critical. One very nice feature of Jovis is its ability to embed an architectural file within a relational file, permitting linkage of relational records to architectural indices. As with the relational commands, there is a series of tutorials that teaches the user how to manipulate flat file constructs in the Jovis environment. The tutorial specifically teaches how to use architectural and relational commands and constructs together, which for many developers may be the heart of the product.

USING JOVIS

Jovis can be used in any XCMD compatible environment. I tested Jovis in a somewhat older version of HyperCard as well as the current version. It also will run in SuperCard, Director, and Oracle Media Objects. Jovis has an initial startup memory requirement of 512K over and above those of the scripting environment. In some

cases, the larger the database the greater the memory requirement.

I found the product difficult to benchmark in a meaningful way. Particularly in a multimedia setting, with various scripting environments and different hardware configurations, any comparison might well be irrelevant to the user/programmer. My overall impression was that performance was more than adequate.

Use of the product requires that the programmer follow some conventions regarding the need for the programmer to create several global variables which would then be for the use of the database engine rather than the stack.

One very nice feature of this product is very good error management. In addition to the XFCN managing and communicating errors, it is possible for your script to intercept and handle error conditions as well. The ability to provide more than "plain vanilla" dialogs reporting an error condition with graceful recovery is critical to a commercial product.

It was difficult to pick a representative complete script to give a flavor of how the product is used. In the end, I simply took the first one in the tutorial, demonstrating creation of a relational file:

```
function CreateNewFile
  global myDB, JovisErrorCode
  —myDB is reserved for exclusive use by Jovis
  if myDB = empty then
    put "Jovis" into myDB
  —initialize myDB
  get Jovis("CreateCollection", "myDB")
  —all calls to Jovis are in the format get Jovis(),
  —with the first parameter the action selector
  if item 1 of JovisErrorCode = "error" then
    answer JovisErrorCode
    put empty into myDB
    return "false"
  else if item 1 of JovisErrorCode = "Warning" then
    answer JovisErrorCode
    put empty into myDB
    return "false"
  end if
  else
    return "false"
  end if
  return "true"
end CreateNewFile
```

Here are some other brief examples:

Create a relation named customers in the file myDB:

```
on CreateDemoRelation
  get Jovis("CreateRelation", "myDB", "Customers")
end CreateDemoRelation
```

Create an text index of 8 character length named Last_Name in the relation Customers in the file "myDB"

```
on CreateTextIndex
  get Jovis("CreateIndex", "myDB", "Customers",
    "Last_Name", "text", 8)
end CreateTextIndex
```

Perform a selection searching on Last_Name in our hypothetical Customers database:

```
on DemoSelect
  ask "Enter last name to set selection to:"
  if the result = "cancel" then exit DemoScript
  put it into Last_Name
```



```

Put "Field Last_Name = [& Last_Name &]" into Criteria
Put "Last_Name, phone, Customer_ID, Account_Start"
    into FieldList
get Jovis("Set Selection", "myDB", "Customers",
    fieldList, Criteria)
end DemoSelect

```

While I have not commented this code, I think the general sense of the steps and syntax is apparent.

LICENSING AND COSTS

This product has a somewhat complex licensing arrangement. The basic price for a single user license (single computer for non-commercial distribution) is \$345. Thus, the person creating a database for personal use is spared the expense of a full commercial license. If the database will be used on more than a single computer, an additional fee of \$350 is required. This additional fee covers everything from unlimited commercial distribution to simultaneous use of the single user product on a second computer in the same office. Hypothetically, a knowledge base with a HyperCard front end and a Jovis database that sells 1000 copies still only costs you \$695 in license fees to DAS Works. Not too shabby.


The server version is more expensive, but again within reason. The basic multiuser package comes with a server and a three user license and costs \$435. Each additional user costs \$85. The server product is protected by a hardware ADB lock. Licenses for the commercial distribution of a multiuser product developed with Jovis are handled on a case by case basis.

THE BOTTOM LINE

Jovis is a relational and flat file engine that lives in the XCMD world. It has a complete set of relational commands nicely complemented by architectural commands. The two schemes can be used together to create traditional relations supplemented by BLOBs: a perfect combination to implement multimedia presentations. Although difficult to judge, it appears that performance is more than adequate. The pricing scheme has many steps but is fair. Were I developing multimedia on Macintosh, I would give this product serious consideration to implement my work.

It would be very helpful if DAS Works created a "User Manual." The reference manual and tutorial are both quite good. However, once the user is acquainted with the product, he or she will want a document on putting it all together without needing to browse the tutorials again, and something more cohesive than a function call reference. Additionally, Jovis is sufficiently different in structure and nomenclature from SQL that some in depth solutions to complex examples are warranted. Finally, I respectfully suggest a set of text files containing the scripts for each of the tutorials.

PRODUCTS REVIEWED

Jovis Single User SDK version 1.04. DAS Works, Inc. 250 West 104th St, Suite 84. New York, NY 10025-4292. Toll free (800) 972-2483. Fax (212) 663-4503. email to <info@dasworks.com> or <sales@dasworks.com>. There is a website at <http://www.dasworks.com/>. 

HELP MAKE MACTECH WORK

Here at *MacTech Magazine*, we rely heavily on outside writers for most of the material that appears in our pages. If readers did not participate in the magazine, sending us their ideas and taking the time to write articles, there would be no *MacTech*. *MacTech Magazine* is not a staff of writers sending a constant stream of one-way messages outwards; it's a living, evolving network of readers conversing with one another, educating one another, sharing their knowledge, their experience, their interest, their trials and tribulations and joys and successes in the constantly unfolding story of programming the Macintosh. *MacTech Magazine* doesn't just happen: it's what the community makes it. If we carry reports of future trends and technologies, if we teach

useful methods, if we review new books and tools, if we provoke thought, provide help, ride the wave of current interests and concerns, it is only because we reflect the thoughts of our readers, who speak through our pages.

You are invited to involve yourself in this exciting conversation amongst readers. No matter who you are, no matter what your credentials may be, if you have a tale to tell, a trick to share, a technique to teach, we want you to consider joining the family of those who write for *MacTech*.

Don't just wait for a topic to be covered or a technique explained in *MacTech*! Take responsibility! Write us an article yourself!

To write for *MacTech*, just send for our Writer's Kit. It's a Microsoft Word file

containing the Styles you need to use, and giving lots of helpful advice and information, including all the legal stuff. You can let us know what you're writing about, or, if you want to, you can just write the article and spring it on us when it's done. [Note: We also have a need for people willing to make themselves available to write occasional product/book reviews.] If we publish your article, you'll be paid for it!

Write to us, the editorial staff, at editorial@mactech.com (or one of the other addresses listed on page 2 of the magazine). Take the future of *MacTech Magazine* into your own hands!

For Macintosh
Programmers & Developers
MacTech
M A G A Z I N E

by Michael Rutman, independent consultant

Macworld Developer Tools Roundup

What's new on the tools for developers

MACWORLD AND THE DEVELOPERS' NEEDS

It's always good to see new products at Macworld, and this year was no exception. This year, there were some concerns about whether there would be any new or exciting things in the developer tools market. The tool developers pleasantly surprised everyone by showing a whole slew of new tools as well as new versions of old tools.

A few companies canceled their products claiming lack of sales, but other companies quickly jumped into the void. The new companies have gone far beyond what the old companies ever tried to accomplish. Most likely, those companies that are now gone actually lost sales to competition rather than a shrinking Macintosh market.

DEVELOPER TOOLS

The core developer's package these days is Metrowerks CodeWarrior. Metrowerks is now shipping CodeWarrior Pro 2.0, as well as a large number of subsidiary products, such as SourceSafe and Latitude. CodeWarrior is their main development IDE with their compilers, and for most Macintosh developers, CodeWarrior Pro 2.0 is a must buy.

SourceSafe is a cross-platform source control system. It integrates with CodeWarrior, and is supposed to be easy to use. Metrowerks started with Microsoft's source code, and they have been bringing it to the Macintosh for several years, giving it stability and a Macintosh user experience.

Latitude is a library for porting Macintosh applications to Rhapsody. For developers with an existing code base that want to live in a Rhapsody world, Latitude is a must buy product. Metrowerks is using Latitude to port their IDE to Rhapsody, so Latitude is getting a major workout.

In addition to Macintosh tools, Metrowerks also provides development environments for creating applications on other platforms, such as BeOS, PalmPilot, and embedded controllers. I've been told there are now 10 back ends for the Metrowerks IDE.

In addition to Metrowerks IDE, another must buy tool is Resorcerer from Mathemaesthetics. Resorcerer is the prime resource-editing tool. For many years, ResEdit, being free, was the premier tool. With Resorcerer 2.0's new features, and ResEdit's lack of support, most developers will now switch over to Resorcerer, despite its very high price. At \$256 a copy, Resorcerer is currently one of the most expensive developer tools on the market.

Resorcerer has always been a resource editor, but done with the Macintosh look and feel. Creating an ALRT (alert) resource didn't just create a resource, it created the entire family of resources alerts need (actb, ALRT, DITL, DLGX, icth). Each resource created is linked together, and all of them can be edited graphically. Non-graphical resources, such as string lists, are edited using a template. New templates are easy to use, and ResEdit templates can be imported.

Resorcerer 2.0 goes beyond resources and includes the ability to edit any type of file as a container. Originally planned for Copland, Resorcerer moves away from the

Michael Rutman is an independent software developer for Macintosh and NeXTSTEP. While working at Software Ventures, he lead the development of Snatcher and MicroPhone Pro for NeXTSTEP. He also worked on the MicroPhone Pro for Macintosh product line. To contact Michael Rutman send mail to moose@manicmoose.com.

resource fork nature of Macintosh files, and instead treats all forks like a container that can hold resources. At first, this may sound like a move away from where its power is, but in reality, it extends the usefulness of the tool. It is now possible to put resources in the data fork. As strange as this sounds, there have been many programs I've worked on where we have embedded resource-like information in the data fork. With Resorcerer's template ability, we would have been able to edit this information graphically.

Another nice product shown was AppMaker version 9. AppMaker's followers will swear by it and refuse to use any interface-creating tool. AppMaker is a bit like Metrowerks' Constructor in that both create the user interface's applications display. Where they differ is in both their power and their ease of use. The easiest way to describe AppMaker is to compare it to Constructor and show where AppMaker shines.

The biggest failing of Metrowerks' Constructor is that its ppob holds only half the information needed to create an interface. Custom views need to be coded in a source file, and if there is any disagreement between the code and the ppob, the window won't display correctly. Furthermore, controls have to be hooked up to actions programmatically. There is no way to test an interface without writing code.

With AppMaker, when an interface is done, source files are generated to produce the desired user interface. The source files produced can be Pascal, C, C++, PowerPlant, and others. The source generated includes all the code for creating and hooking up views, including custom views. This automation prevents many errors that creep into CodeWarrior projects that use Constructor.

Another nice feature of AppMaker is its separation of user interface and back end. AppMaker creates source files that control the user interface, and creates stub files to talk to the source files created. With this model, a change in the UI does not break the underlying code. This model also helps prevent the proliferation of objects that do nothing more than tell PowerPlant which window is being loaded.

The demo of AppMaker was a Fahrenheit to Celsius converter. It consisted of a window with 2 text fields, a slider, and a thermometer. Using the inspector in AppMaker, controls were hooked up, formulas were created, and everything was done. Four lines of code, selecting create code, a compile, and it worked. No interaction with the source files was necessary at all. When asked how hard it would be to programmatically change the values, I was told that all the variables have public accessor methods, and calling those will change the values everywhere.

The next type of developer tool everybody has need for is an installer. The installer market has been going through a shakeup over the last few years with large numbers of companies going from Aladdin's InstallerMaker to MindVision's VISE. Both Aladdin and MindVision have new versions, and both respond to fundamental needs of their customers. A comparison between the two products is always tricky, but both offer limited versions on their web site to allow everyone a chance to see for themselves.

Aladdin's biggest customer problem was speed and ease of use. Earlier versions of InstallerMaker could not hold up in a comparison against VISE, and InstallerMaker sales suffered for it. InstallerMaker uses the tried and true Stuffit Engine to get good compression reducing the size of the packages for installation. The new version of InstallerMaker adds installing from ftp sites, uninstalling, time-limited demos, and updater files. Overall, it looks like Aladdin is finally responding to MindVision's growing market share.

In the last year or two, MindVision grabbed the lead with a clearly superior product. Their latest version capitalized on their lead to add a slew of new features. VISE now has hierarchical packages, multiple targets from a single script, custom layout, better drag and drop support, better billboard support, web installer, and many other new features. VISE 5.0 has an improved user interface that many developers will find even easier to use.

Another must have tool is Bare Bones software's BBEdit 4.5. BBEdit has always been a useful tool, and it just gets better. Many Unix users show off the power of emacs and say the Macintosh is just a toy. I can always respond by launching BBEdit and saying emacs is just a kludge. BBEdit is the text editing tool for the Macintosh that puts any other text editor on any other platform to shame. If I started listing all the features of BBEdit now, I would not finish before they came out with a new version with even more features in it. Not only is BBEdit a powerful and useful tool, its relatively cheap price makes it a must buy product.

Altura software has a new version of ObjectMaster. ObjectMaster understands C++, Objective-C, and Java objects, and allows browsing the objects. While other editors, such as Metrowerks and BBEdit, allow some class browsing, ObjectMaster is built around it. A large number of features make ObjectMaster a very useful tool to have around. ObjectMaster does not support compiling a program, but will seamlessly talk to most IDE's, including Metrowerks and MPW. Altura has done a good job of creating a development environment where you code like you design.

The last developer tool is a different type of "tool". Developer Depot, however, is genuinely a useful tool to have in that they have all the developer tools for sale at a good price (with lowest price guarantee). In addition, they help sponsor the pavilions and advertising for other developer tools. For a developer wanting to buy one or more of the products I've described, Developer Depot is a must contact company. [Disclaimer: While both Developer Depot and MacTech Magazine are run very separately, they are owned by the same parent company.]

MacTech also has a CD with all of its issue through May 1997 fully indexed. Anyone who has let their subscriptions lapse for a year or two, should pick up the CD and see what they missed.

RHAPSODY DEVELOPMENT TOOLS

There were a few Rhapsody development tools at the show, but the only one I found useful for Macintosh developers right now is Joy. There are other useful and cool Rhapsody tools coming soon, but Joy was shown at Macworld. The demo of Joy is overwhelming, and a full review would be hard pressed to do this product justice, but I believe that Joy takes development tools to a new level.

Basically, Joy is a command line interface, which can be added to your application. The technology behind the command line relies on Rhapsody's dynamic method dispatch mechanism. Any object can be accessed, and any method can be called. As everything is done at run-time, any method with any parameters can be called on any object in any application linked to Joy.

While an interactive command line to an application is a powerful tool, the creators of Joy have gone way beyond what I've described. Hopefully, at a later time, we can present a full review of Joy to our readers.

OTHER TOOLS DEVELOPERS WOULD FIND USEFUL

While developers will use most productivity software, there are some products, such as product management software, that are of interest to developers even though they are not developer-only tools. Walking the show floor, I found quite a few new products that many developers will find a useful addition.

One of the current dilemmas facing Macintosh programmers today is cross-platform connectivity. AppleShare is a great tool for transferring files between Macintoshes, but when Windows machines are involved, something extra is needed. Two products resolve this, Miramar's PC MacLan, and Dave. Both products allow file transfer and printing between the platforms, but they do it from the opposite sides. Dave is installed on the Macintosh and runs on the Macintosh talking Windows. MacLan installs on a Windows machine with Macintosh clients talking their own language. Depending on your situation, either product may be more appropriate. As a Macintosh user, I found MacLan difficult to install, but I'm sure that anyone who knows Windows would find it easy.


Dave, on the other hand, installs and acts like a Macintosh program should. Without reading any installation directions, I installed Dave, and in the Apple Menu, there was a DAVE Access menu item, exactly like I expected. Opening it told me that I had not configured NetBIOS, and told me where to go to configure NetBIOS. From the NetBIOS control panel, as I moved the mouse around, each field was described below; a kind of contextual help that made configuring easy. The only down side to Dave was my needing to go to the Windows machine and set up domains and stuff.

Another useful tool I saw was PageCharmer. PageCharmer is a set of useful Java widgets ready for customization and insertion into a web site. Most developers I know have a web site, and making a cool web site can drain hundreds of hours. There are many web page creation tools,

from DreamWeaver to BBEdit, but PageCharmer isn't for creating web pages, it's for adding the widgets to make a page look nice. It has a very nice interface, and customizing buttons and maps was very enjoyable.

While not a tool, I did look at the Contour mouse. The Contour mouse is molded to fit the hand, and different size mice are available. As I suffer from a bit of RSI, I am always looking for new ergonomic devices. The mouse has a different feel from the trackball and normal mice. I don't feel the need to grip it as hard as a normal mouse, but I do feel it a bit more in the elbow. I haven't had the mouse long enough to know if it works better or not, but I suspect that it will help my wrist, where I need it most. The mouse does have two drawbacks, and both of those are only relevant if you share a machine. The mouse, being sized and molded, will only fit my middle-sized right hand. If my wife, who is left-handed and has a smaller hand, wants to use my machine, she will have trouble.

CONCLUSIONS

Quite a number of new and updated products were all over the show floor. A lot of good hard work has gone into making developers' lives easy. For those that could not attend the show, I hope I've been able to bring you just a taste of the tools that can make your programming easier. 

**Want to have the
entire history
of MacTech Magazine at
your fingertips?**

**Check out the MacTech
CD-ROM with superfast
searching.**

For more info, see

<<http://www.mactech.com>>

or write

<<mailto:custservice@devdepot.com>>

by Andrew Downs

Java Serialization

Adding object persistence to Java applications

INTRODUCTION

This article discusses and demonstrates how to incorporate object persistence into a Java application using the serialization mechanism in Java 1.1. This article assumes a general familiarity with Java. The code in this article was developed using the Apple Macintosh Runtime for Java (MRJ) version 2.0 and the MRJ SDK.

OVERVIEW

Serialization involves saving the current state of an object to a stream, and restoring an equivalent object from that stream. The stream functions as a container for the object. Its contents include a partial representation of the object's internal structure, including variable types, names, and values. The container may be transient (RAM-based) or persistent (disk-based). A transient container may be used to prepare an object for transmission from one computer to another. A persistent container, such as a file on disk, allows storage of the object after the current session is finished. In both cases the information stored in the container can

later be used to construct an equivalent object containing the same data as the original. The example code in this article will focus on persistence.

Since Java applets do not have direct access to a local disk, it may be impossible for an applet to find a suitable container for persistent storage of a serialized object. Therefore, the code in this article focuses on Java applications.

IMPLEMENTATION

For an object to be serialized, it must be an instance of a class that implements either the `Serializable` or `Externalizable` interface. Both interfaces only permit the saving of data associated with an object's variables. They depend on the class definition being available to the Java Virtual Machine at reconstruction time in order to construct the object.

The `Serializable` interface relies on the Java runtime default mechanism to save an object's state. Writing an object is done via the `writeObject()` method in the `ObjectOutputStream` class (or the `ObjectOutput` interface). Writing a primitive value may be done through the appropriate `write<datatype>()` method. Reading the serialized object is accomplished using the `readObject()` method of the `ObjectInputStream` class, and primitives may be read using the various `read<datatype>()` methods.

What about other objects that may be referred to by the object we are serializing? For instance, what if our object is a `Frame` containing a set of (AWT) `Panel` and `TextArea` instance variables? Using the `Serializable` interface, these references (and their associated data) also are converted and written to the stream. All state information necessary to reconstruct our `Frame` object and any objects that it references gets stored together.

Andrew Downs is a Senior Software Engineer for Template Software in New Orleans, LA, designing and building enterprise apps. He also teaches C and Java programming at Tulane University College. Andrew wrote the Macintosh freeware program *Recent Additions*, and the Java application *UDPing*. You can reach him at andrew.downs@template.com.

If those other objects or their formats weren't stored, our reconstructed Frame would contain null object references, and the content of those Panels and TextAreas would be gone. Plus, any methods that rely on the existence of the Panels or TextAreas would throw exceptions.

The Externalizable interface specifies that the implementing class will handle the serialization on its own, instead of relying on the default runtime mechanism. This includes which fields get written (and read), and in what order. The class must define a `writeExternal()` method to write out the stream, and a corresponding `readExternal()` method to read the stream. Inside of these methods the class calls `ObjectOutputStream writeObject()`, `ObjectInputStream readObject()`, and any necessary `write<datatype>()` and `read<datatype>()` methods, for the desired fields.

HIDING DATA

Sometimes you may wish to prevent certain fields from being stored in the serialized object. The Serializable interface allows the implementing class to specify that some of its fields do not get saved or restored. This is accomplished by placing the keyword *transient* before the data type in the variable declaration. For example, you may have some data which is confidential and can be re-read from a master file later (as opposed to saving it with the serialized object). Or you decide (wisely) to preserve the privacy of file references by declaring any such variables as transient. Otherwise, all fields automatically get written without any additional effort by the class.

In addition to those fields declared as transient, static fields are not serialized (written out), and so cannot be deserialized (read back in).

Another way to use Serializable, and control which fields get written, is to override the `writeObject()` method of the Serializable interface. Inside of this method, you are responsible for writing out the appropriate fields. If you take this approach, you will want to override `readObject()` as well, to control the restoration process. This is similar to using Externalizable, except that interface requires `writeExternal()` and `readExternal()`.

For the Externalizable interface, since both `writeExternal()` and `readExternal()` must be declared public, this increases the risk that a rogue object could use them to determine the format of the serialized object. For this reason, you should be careful when saving object data with this interface.

It is worth considering the amount of security you need for any objects that you serialize. When reading them back in, all of the normal Java security checks (such as the bytecode verifier) are in effect. You can define certain values within the class that should remain intact in serialized objects. Perhaps they should contain a specific value, or a value within a particular range. You can easily check the value of any

numeric variable read in from a serialized object, especially if you know that only a portion of the available range for that data type is used by your variable.

You can also encrypt the outgoing data stream. The implementation is up to you, and don't forget to decrypt the object format when reading it back in.

VERSIONING

The ability to save and restore objects leads to an interesting question: what happens when an object has been stored for so long, that upon restoration it finds that its format has been superseded by a new, different version of the class?

The stream reading the serialized representation is responsible for accounting for any differences. The intent is that a newer version of a Java class should be able to interoperate with older representations of the same class, as long as there have not been certain changes in the class structure. The same does not necessarily hold true for an older version of the class, which may not be able to effectively deal with a newer representation.

So, we need some way to determine at runtime (or more appropriately, deserialization-time) whether we have the necessary backward compatibility.

In Java 1.1, changes to classes may be specified using a version number. A specific class variable, `serialVersionUID` (representing the Stream Unique Identifier, or SUID), may be used to specify the earliest version of the class that can be deserialized. The SUID is declared as follows:

```
static final long serialVersionUID = 2L;
```

This particular declaration and assignment specifies that version 2 is as far back as this class can go. It is not compatible with an object written by version 1 of the class, and it cannot write a version 1 object. If it encounters a version 1 object in a stream (such as when restoring from a file), an `InvalidClassException` will be thrown.

The SUID is a measure of backward compatibility. The same SUID can be used for multiple representations of a class, as long as newer versions can still read the older versions.

If you do not explicitly assign a SUID, a default value will be assigned when the object gets serialized. This default SUID is a hash, or unique numeric value, which is computed using the class name, interfaces, methods, and fields. The exact algorithm is defined by the Secure Hash Algorithm (SHA). Refer to the Sun Java documentation for details.

The JDK (MRJ) utility program `serialver` will display the default (hash) SUID for a class. You can then paste this value in any subsequent, compatible versions of the class. (It is not required in the initial version of the class.) As of this writing the `serialver` program has not been included in the MRJ SDK, but hopefully will be in the future.

continued on page 66

by Bob Boonstra, Westford, MA



MANCALA

A stocking stuffer from this past Christmas provided the inspiration for this month's Challenge. Santa gave me a two player travel game called Mancala which might provide some amusement the next time I travel by car or by plane, provided the ride is smooth enough to keep the small stones inside the bowls of the game board. I thought it might make for an interesting Challenge tournament.

The basic Mancala game consists of a board with 14 hollowed-out bowls arranged in an oval form, one large bowl at each end of the board, and six smaller bowls facing each of two players seated opposite one another. Each player "owns" the large bowl, or "mancala", positioned to his right, and the six small bowls closest to him. The game starts with each small bowl containing four stones. The game begins with the first player picking up all of the stones in one of his small bowls, dropping one stone in the bowl to the right, the second stone in the second bowl on the right, continuing around the board in counterclockwise fashion until the stones he picked up are gone. The second player then picks up the stones in one of his small bowls, drops them one at a time in the bowls to the right, etc. The game ends when one player cannot move (i.e., no stones remain in that player's small bowls). The winner is the player with the most stones in his mancala. There are a number of variations to the game, and the specific restrictions on our Mancala Challenge tournament are explained below.

The prototype for the code you should write is:

```
#if defined(__cplusplus)
extern "C" {
#endif

Boolean Mancala(
    long board[],           /* on entry, board[i] is number of stones in bowl i */
                           /* on exit, board reflects the results of your move */
    const long boardSize, /* number of bowls in the board, including mancalas */
    void *privStorage,    /* pointer to 1MB of storage for your use */
    const Boolean newGame, /* true for your first move of a game */
    const Boolean playerOne, /* true when you are the first player */
    long *bowlPlayed,      /* return the number of the bowl you played from */
    long *directionPlayed /* return 1 if you played counter-clockwise, */
                           /* return -1 if you played clockwise */
);

#ifdef __cplusplus
}
#endif
```

Each time your Mancala routine is called, you will be provided with a `board[]` array that indicates the number of stones in each bowl, including the Mancalas, at the beginning of your turn. The `boardSize` parameter will indicate the number of bowls in the board – in the standard Mancala game, this would be 14, but in our Challenge it might be any even number between 8 and 32, inclusive. The mancala for the first player will be `board[0]`, while the mancala for the second player will be `board[boardSize/2]`. You will also be provided a pointer `privStorage` to 1MB of storage, preinitialized to zero, for each of your moves in a single game. For your first move of a game, `newGame` will be TRUE, otherwise `newGame` will be false. If you are the first player, `playerOne` will be TRUE for each of your

THE RULES

Here's how it works: each month we present a new programming challenge. First, write some code that solves the challenge. Second, optimize your code (a lot). Then, submit your solution to MacTech Magazine. We choose a winner based on code correctness, speed, size, and elegance (in that order of importance) as well as the submission date. In the event of multiple equally desirable solutions, we'll choose one winner (with honorable mention, but no prize, given to the runner up). The prize for each month's best solution is a \$100 credit for Developer Depot™.

Unless stated otherwise in the problem statement, the following rules apply: All solutions must be in ANSI compatible C or C++, or in Pascal. We disqualify entries with any assembly in them (except for challenges specifically stating otherwise.) You may call any Macintosh Toolbox routine (e.g., it doesn't matter if you use `NewPtr` instead of `malloc`). We compile all entries into native PowerPC code with compiler options set to enable all available speed optimizations. The development environment to be used for selecting the winner will be stated in the problem. **Limit your code to 60 characters per line** or compress and binhex the

solution; this helps with e-mail gateways and page layout.

We publish the solution and winners for each month's Programmer's Challenge three months later. All submissions must be received by the 1st day of the month printed on the front cover of this issue.

You can get a head start on the Challenge by reading the Programmer's Challenge mailing list. It will be posted to the list on or before the 12th of the preceding month. To join, send an email to listserv@listmail.xplain.com with the subject "subscribe challenge-A".

Mark solutions "Attn: Programmer's Challenge Solution" and send it by e-mail to one of the Programmer's Challenge addresses in the "How to Communicate With Us" section on page 2 of this issue. Include the solution, all related files, and your contact info.

MacTech Magazine reserves the right to publish any solution entered in the Programmer's Challenge. Authors grant MacTech Magazine the exclusive right to publish entries without limitation upon submission of each entry. Authors retain copyrights for the code.

moves, otherwise `playerOne` will be `FALSE`. You should return the index of the bowl you played from in `bowlPlayed`, and you should return the direction you chose to move in `directionPlayed`. You should also update your view of the number of stones in each bowl in `board[]`.

There are a number of rule variations for Mancala. We will play with the following additions to the standard rules:

- the `board` will contain between eight and 32 bowls, instead of the standard 14.
- at the beginning of the game, each small bowl will have between two and 16 stones (instead of the standard four), with the same number in each bowl.
- players do not drop stones into their opponent's mancala.
- players may choose to move either counter-clockwise or clockwise on a given move.
- if a player drops the last stone into his mancala, he gets to move again (my test code will call your Mancala routine again).
- if a player drops the last stone into one of the empty bowls (`board[i]`) on his side of the `board`, he takes that stone, plus all the stones in his opponent's bowl directly across from his bowl (`board[boardSize-i]`) and places them in his mancala.
- the game ends when one player has no stones in any of his small bowls and cannot move. The other player then places all remaining stones from his small bowls into his mancala.

At the end of the game, each player will be credited with one point for each stone in his mancala, minus one point for each 100ms of cumulative execution time. The Challenge winner will be the entry that accumulates the most points in a round-robin tournament where each entry competes against each other entry twice for each set of game parameters, once playing first, and once playing second.

This will be a native PowerPC Challenge, using the latest CodeWarrior environment. Solutions may be coded in C, C++, or Pascal.

THREE MONTHS AGO WINNER

Congratulations once again to **Ernst Munter** (Kanata, Ontario) for submitting the fastest entry to the January Cell Selection Challenge. Readers were invited to implement a C++ `CellSelection` class, including methods to add one selection to another, remove one selection from another, invert a selection, count the number of active cells in a selection, and determine if two selections were equal. A dual-processor 8500/2x200 was used to test this Challenge, and contestants were free to take advantage of the multiple processors, either through the Apple/Daystar Multiprocessing API, or by programming for BeOS and taking advantage of the symmetric multiprocessing features of that operating system.

Or so I thought when I structured the problem. As it turned out, of the four solutions submitted, only one took advantage of the multiprocessor opportunity. I'll discuss the use of multiprocessing in that solution later in the article.

Ernst based his solution on the "vixel" data structure he used to win the Intersecting Rectangle Challenge of two years ago. His code is well-commented, so I'll let his solution speak for itself, except to point out that one other entry made reference to the "vixel" approach. It's nice to see readers put winning code from the past to good (and efficient) use.

The table below lists the total execution time in milliseconds for all test cases, as well as execution times for five individual test cases. It also lists code size, data size, and the number of processors used for each entry. The number in parentheses after the entrant's name is the total number of Challenge points earned in all Challenges to date prior to this one. The entries marked with an asterisk are those which did not complete one or more test cases. Test times in italics are extrapolated from partial test cases that were successfully completed. (See table 1).

Ulf Schröder's entry was the only one to attempt to use both of the processors available in the test machine. I tested a version of Ulf's entry modified to use only one processor, but the results were not much different, suggesting that either the `CellSelection` problem is not amenable to partitioning, or that Ulf's technique for doing so didn't have much affect for the particular test cases I used.

This Challenge was intended to be about multiprocessing,

Table 1

Name	Total	Execution Time (msecs)					Code	Data	#
		Case1	Case2	Case3	Case4	Case5	Size	Size	Proc
Ernst Munter (310)	1040	166	134	144	293	303	20352	5314	1
Willeke Rieken (10)	19114	185	255	267	8	18399	21784	1292	1
Ulf Schröder (*)	10532	287	104	92	108	9939	11276	2092	2
R. B. (*)	—	—	—	—	—	—	10612	1226	1

so it is worth spending a minute on how Ulf tried to take advantage of it. The first step was to determine how many processors were available and to create a task entry and a pair of semaphores for each processor:

```
if (!MPLibraryIsLoaded())
    gProcessors = 1;
else
    gProcessors = MPProcessors();

// Create the tasks if more than one processor
if (gProcessors > 1)
{
    OSErr err;
    .....
    for (int32 i = 0; i < gProcessors - 1; ++i)
    {
        err = MPCreateSemaphore(1, 0,
            &gTaskData[i].mStartSemaphore);
        if (err != noErr) throw err;

        err = MPCreateSemaphore(1, 0,
            &gTaskData[i].mFinishedSemaphore);
        if (err != noErr) throw err;

        err = MPCreateTask(
            task, &gTaskData[i], 0, gTerminationQueue,
            0, 0, 0, &gTaskData[i].mTaskId);
        if (err != noErr) throw err;
    }
}
```

This step is done once, at initialization time. Then later, when the code determines that it has a problem worth partitioning, it divides the problem, assigns one piece to another task (or tasks, in the general case), signals the other task to begin work, solves the remaining piece of the problem, and then waits for the other task to complete its portion.

```
if (gProcessors > 1 && length > kMinLengthForMP)
{
    uint32 mid = length / 2;
    .....
    DoRemove doRemove(area, restEnd);
    DoRemove doRemove1(area, rest1End);

    gTaskData->mBegin = mAreas.begin() + mid;
    gTaskData->mEnd = mAreas.end();
    gTaskData->mNewEnd = gTaskData->mEnd;
    gTaskData->mFunction = &doRemove1;

    // Start the other task
    MPSignalSemaphore(gTaskData->mStartSemaphore);

    // Do my part of the job
    AreaList::iterator newEnd =
        remove_if(
            mAreas.begin(),
            mAreas.begin() + mid,
            doRemove);

    //Wait for other task
    MPWaitOnSemaphore(gTaskData->mFinishedSemaphore,
        kDurationForever);
    .....
}
```

For further information, there is an introduction to multiprocessing on the Mac in the March, 1996, issue of MacTech Magazine, or you can visit Apple's web site to obtain MultiProcessing API in the SDK, including some sample code.

TOP 20 CONTESTANTS

Here are the Top Contestants for the Programmer's Challenge, including everyone who has accumulated more than 10 points during the past two years. The numbers below include points awarded over the 24 most recent contests, including points earned by this month's entrants.

Rank	Name	Points	Rank	Name	Points
1.	Munter, Ernst	218	10.	Day, Mark	20
2.	Boring, Randy	73	11.	Higgins, Charles	20
3.	Cooper, Greg	61	12.	Hostetter, Mat	20
4.	Lewis, Peter	51	13.	Rieken, Willeke	20
5.	Mallett, Jeff	50	14.	Studer, Thomas	20
6.	Nicolle, Ludovic	44	15.	Hart, Alan	14
7.	Murphy, ACC	34	16.	O'Connor, Turlough	14
8.	Gregg, Xan	28	17.	Picao, Miguel Cruz	14
9.	Antoniewicz, Andy	24			

There are three ways to earn points: (1) scoring in the top 5 of any Challenge, (2) being the first person to find a bug in a published winning solution or, (3) being the first person to suggest a Challenge that I use. The points you can win are:

1st place	20 points	5th place	2 points
2nd place	10 points	finding bug	2 points
3rd place	7 points	suggesting Challenge...	2 points
4th place	4 points		

Here is Ernst's winning solution to the CellSelection Challenge:

Cells.h

© 1998 Ernst Munter

```
/*
    "Cell Selection"

```

This file is a C++ header file containing the CellSelection class, as well as a number of auxiliary structs and classes.

The purpose of the CellSelection class is to contain a set of cells in a 2-D world. Cells can be added and manipulated in groups represented as sets of cells, called Areas.

Solution Strategy

The overall idea is the same as the one I used in the "Intersecting Rectangles"

challenge (solution in MacTech of Apr 96).

Cells are similar to black-and-white pixels, and any area of all-black (off) or all-white (on) pixels can be represented by a single bit and the 4 edge coordinates. I had called such a group of pixels a "vixel".

The CellSelection class maintains two sets of coordinates, rows and columns, and a bit map of vixels.

The empty selection starts out with a single (off) vixel, covering the area defined by the range of 32-bit integers (left=2147483648, top=2147483648, right=2147483647, bottom=2147483647).

As areas are added or removed, missing coordinates are inserted and the total area becomes further and further divided into sub-areas.

Coordinate sets are stored in trees to facilitate easy lookup. In addition, they are linked in linear lists to make it easier to traverse a span of coordinates corresponding to the edges of a given area which may be represented by a rectangular block of many sub-areas of vixels.

Memory Management

As row and column coordinates are added, each is allocated by "new". Each row also holds a pointer to "vixels" which are allocated with each row as needed. The default amount of vixel bits per row is set to 512. If more vixels are needed as the CellSelection grows, rows are expanded in increments of 512 bits. The bit map is not sorted right to left; rather, each column struct contains an index to the bit-position in the vixel array. The bits are allocated from the vixels arrays in the order of arrival with the method calls.

All calls to "new" are in try/catch blocks in order to allow the test program to fail gracefully if we should run out of memory.

Optimisations

Basically none.

Memory, rows and columns, get allocated as needed, but are only deleted when the CellSelection is destroyed. This will probably slow things down as the vixel map gets more and more fragmented in large selections.

A future improvement could be a garbage collector which detects when adjacent rows or columns are identical and can be merged.

Alternatively, CellSelections could be kept in a "normal" state during each operation by avoiding unnecessary splits of coordinates, and detection of pairs that can be merged.

Extras

I have added some public methods to the CellSelection class to simplify testing:

```
int CellSelection::Width()
int CellSelection::Height()
    return the width and height of the vixel map

void CellSelection::ResetTestAreas()
bool CellSelection::NextTestArea(Area* a)
    permit stepping through the vixel map and identify
    each sub-area that contains a block of cells
```

I have also added a method to the Area struct to test for empty areas according to the definition.

Clarifications

Methods with return type "bool" (except EqualSelected()) return false only if we run out of memory. They return true in all other cases, including calls with empty areas.

The class makes no assumptions which could restrict the range of areas. Specifically, an area with a right or bottom coordinate of 2147483647 will be handled correctly, i.e. we do not need a value 1 greater than the highest value coordinate of an area. */

```
#include <string.h>
// needed for memcpy and memset
```

```
#if __dest_os == __be_os
#include <SupportDefs.h>
```

```
#else
typedef long int32;
typedef unsigned long uint32;
#endif
```

```
#define EXTRA_FOR_TESTING 1
```

```
static const enum ProgConsts{
    kIncrementBits=512,
    kNegInfinity=0x80000000L,
    kPosInfinity=0x7FFFFFFFL
}
// the following line avoids a compiler warning in CW-PRO2
Consts(kNegInfinity);
```

Area

```
struct Area {
    int32 left, top, right, bottom;
    /* Area coordinates are inclusive. {2,2,3,4} includes 6 cells. */
    const bool IsEmpty() const {
        /* Any area with left>right or top>bottom is empty. */
        return ((left>right) | (top>bottom));
    }
};
```

Node

```
struct Node {
    int32 lo;
    Node* nextNode;
    Node* leftNode;
    Node* rightNode;
    int bal;
    Node(int32 x) {
        lo=x;
        leftNode=rightNode=nextNode=0;
        bal=0;
    }
    int32 Limit() const {
        return (nextNode?nextNode->lo-1:kPosInfinity;
    }
    uint32 Slice(int32 xlo, int32 xhi) const {
        int32 upperLimit=Limit();
        return 1+(xhi<upperLimit?xhi:upperLimit)-(xlo>lo?xlo:lo);
    }
};
```

Row:Node

```
struct Row:Node{
    int size; // number of bytes allocated
    char* vixels; // 1 bit per vixel
    Row(int32 x):Node(x){
        size=kIncrementBits/8;
        try {vixels=new char[size];}
        catch (...) {vixels=0;return;}
        memset(vixels,0,size);
    }
};
```

**Interested in advertising in
MacTech Magazine?**

**Contact our
ad sales department at
<mailto:adsales@mactech.com>**


```

)
Row(Row* rp):Node(rp->lo){
    size=rp->size;
    try {vixels=new char[size];}
    catch (...) {vixels=0;return;}
    memcpy(vixels,rp->vixels,size);
}
~Row(){delete[] vixels;}
Row* Next() const {return (Row*)(nextNode);}
bool Stretch(uint32 index,uint32 width) {
//if necessary, we replace this row with a wider row
    if (width >= size*8) {
        int newSize=size+kIncrementBits/8;
        char* newVixels;
        try{newVixels=new char[newSize];}
        catch(...) {return false;}
        memcpy(newVixels,vixels,size);
        memset(newVixels+size,0,kIncrementBits/8);
        delete vixels;
        size=newSize;
        vixels=newVixels;
    }
    if (Vixel(index)) SetVixel(width); //else already 0;
    return true;
}
#define MEMBER vixels[index>>3]
#define BIT (1L<<(index&7))
int Vixel(uint32 index) const {return MEMBER & BIT;}
void ClearVixel(uint32 index) {MEMBER &= ~BIT;}
void SetVixel(uint32 index) {MEMBER |= BIT;}
void InvertVixel(uint32 index) {MEMBER ^= BIT;}
#undef MEMBER
#undef BIT
bool InitFailed() const {return (vixels==0);}
};

```

Col:Node

```

struct Col:Node {
    uint32 index;// bit number in every row of vixels
    Col(int32 xLeft,uint32 xIndex):
        Node(xLeft){index=xIndex;}
    Col* Next() const {return (Col*)(nextNode);}
};

```

```

class Tree {
// based on AVL balanced binary tree, see:

```

Tree

```

// "Algorithms and Data Structures in C++"
// by Leendert Ammerdaal, published by Wiley 1996
// Here, the tree is never allowed to be empty,
// so we do not have to check for NULL pointers
private:
    Node* root;
    void LeftRotate(Node* &p) {
        Node* q=p;
        p=p->rightNode;
        q->rightNode=p->leftNode;
        p->leftNode=q;
        q->bal--;
        if (p->bal>0) q->bal=p->bal;
        p->bal--;
        if (q->bal<0) p->bal+=q->bal;
    }
    void RightRotate(Node* &p) {
        Node* q=p;
        p=p->leftNode;
        q->leftNode=p->rightNode;
        p->rightNode=q;
        q->bal++;
        if (p->bal<0) q->bal=p->bal;
        p->bal++;
        if (q->bal>0) p->bal+=q->bal;
    }
    int Insert(Node* &p,Node* q,Node* prev);
    Node* Find(Node* p,int x) const;
public:
    void Clear(){root=0;}
    Node* Root(){return root;}
    void Insert(Node* q,Node* prev) {
        Insert(root,q,prev);
    }
    Node* Find(int x) const {
        if (root) return Find(root,x);
        return root;
    }
};

```

CellSelection

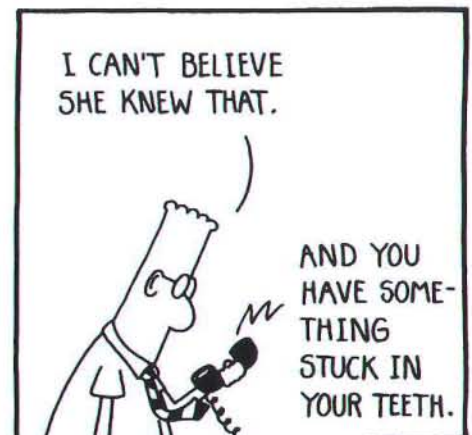
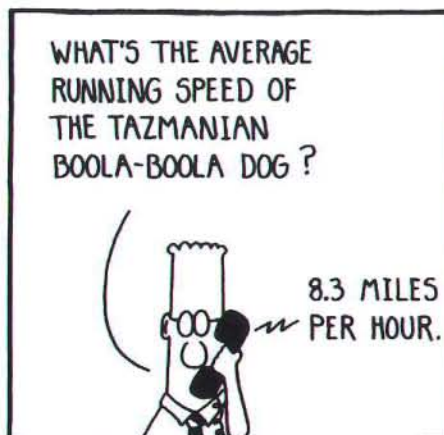
```

class CellSelection {
private:
    Tree rowTree; // tree of sub-areas
    Row* row; // top most row, never deleted
    Tree colTree; // tree of column indices
    Col* col; // left most column, never deleted

```

Dilbert® by Scott Adams

Reprinted by permission on UFS, Inc.



©1997 United Feature Syndicate, Inc. (NYC)


```

uint32 width; // number of columns

#ifdef EXTRA_FOR_TESTING
    Row* testRow;
    Col* testCol;
#endif

bool CreateEmpty() {
    // Creates a single vixel covering the 32-bit universe
    rowTree.Clear();
    colTree.Clear();
    try {row=new Row(kNegInfinity);}
    catch (...) {return false;}

    if (row->InitFailed())
        return false;
    rowTree.Insert(row,0);
    try {col=new Col(kNegInfinity,0);}
    catch (...) {delete row; return false;}
    colTree.Insert(col,0);
    width=1;
    return true;
}

void FreeAll() {
    // Uses linked lists to delete all rows and columns
    Row* rp=row;
    while (rp) {
        Row* nextRow=rp->Next();
        delete rp;
        rp=nextRow;
    }
    Col* cp=col;
    while (cp) {
        Col* nextCol=cp->Next();
        delete cp;
        cp=nextCol;
    }
}

// and clear the trees, in preparation for re-use
rowTree.Clear();
colTree.Clear();
}

//The next three methods expand the vixel map by
//insertion of new rows and columns

Row* SplitRow(Row* rp,int32 newTop) {
    // Creates a new row as a copy of rp, and inserts
    // it after it, with new coordinates marking the split
    Row* newRow;
    try {newRow=new Row(rp);}
    catch (...) {return 0;}
    if (newRow->InitFailed())
        return 0;
    newRow->lo=newTop;
    rowTree.Insert(newRow,rp);
    return newRow;
}

bool StretchRows(uint32 index,uint32 width) {
    // Allocates the next available vixel column
    Row* rp=row;
    while (rp) {
        if (!rp->Stretch(index,width)) {
            // If we run out of memory before stretching all rows
            // we fail. Those rows that did get stretched stay there.
            // No harm done.
            return false;
        }
        rp=rp->Next();
    }
    return true;
}

Col* SplitColumn(Col* cp,int32 newLeft) {
    // Creates a new column as a copy of cp, and inserts
    // it after it, with new coordinates marking the split
    if (!StretchRows(cp->index,width))

```

```

        return 0;
    Col* newCol;
    try {newCol=new Col(newLeft,width);}
    catch (...) {return 0;}
    colTree.Insert(newCol,cp);
    width++;
    return newCol;
}

void OutlineArea(Area a,Col** left,Row** top) const {
    // locate Top-left corner closest to area.
    *top=(Row*)(rowTree.Find(a.top));
    *left=(Col*)(colTree.Find(a.left));
}

bool DefineArea(Area a,Col** left,Row** top) {
    // locate area and create new borders if needed
    Row* rp=(Row*)(rowTree.Find(a.top));
    if (rp->lo == a.top) *top=rp; else
    if (0==(*top=SplitRow(rp,a.top))) return false;

    rp=(Row*)(rowTree.Find(a.bottom+1));
    if (rp->lo != a.bottom+1)
    if (!SplitRow(rp,a.bottom+1)) return false;

    Col* cp=(Col*)(colTree.Find(a.left));
    if (cp->lo == a.left) *left=cp; else
    if (0==(*left=SplitColumn(cp,a.left))) return false;

    cp=(Col*)(colTree.Find(a.right+1));
    if (cp->lo != a.right+1)
    if (!SplitColumn(cp,a.right+1)) return false;

    return true;
}

//The next set of methods scan blocks of vixels and perform
//the indicated function on each vixel
bool SetArea(Area a) {
    Col* left;
    Row* top;
    if (!DefineArea(a,&left,&top)) return false;
    do {
        Col* cp=left;
        do {
            top->SetVixel(cp->index);
            cp=cp->Next();
        } while ((cp) && (a.right>cp->lo));
        top=top->Next();
    } while ((top) && (a.bottom>top->lo));
    return true;
}

bool ClearArea(Area a) {
    Col* left;
    Row* top;

```

**Want to get product
information about a MacOS
development tool? Write
<<mailto:custservice@devdepot.com>>**


```

    if (!DefineArea(a,&left,&top)) return false;
    do {
        Col* cp=left;
        do {
            top->ClearVixel(cp->index);
            cp=cp->Next();
        } while ((cp) && (a.right>=cp->lo));
        top=top->Next();
    } while ((top) && (a.bottom>=top->lo));
    return true;
}

bool InvertArea(Area a) {
    Col* left;
    Row* top;

    if (!DefineArea(a,&left,&top)) return false;
    do {
        Col* cp=left;
        do {
            top->InvertVixel(cp->index);
            cp=cp->Next();
        } while ((cp) && (a.right>=cp->lo));
        top=top->Next();
    } while ((top) && (a.bottom>=top->lo));
    return true;
}

bool AllSelected(Area a,bool test) const {
    Col* left;
    Row* top;
    OutlineArea(a,&left,&top);

    do {
        Col* cp=left;
        do {
            if (test != IsSet(top,cp)) return false;
            cp=cp->Next();
        } while ((cp) && (a.right>=cp->lo));
        top=top->Next();
    } while ((top) && (a.bottom>=top->lo));
    return true;
}

uint32 CountSelectedP(Area a) const {
    Col* left;
    Row* top;
    OutlineArea(a,&left,&top);
    uint32 count=0;

    do {
        Col* cp=left;
        uint32 strip=0;
        do {
            if (IsSet(top,cp)) strip+=cp->Slice(a.left,a.right);
            cp=cp->Next();
        } while ((cp) && (a.right>=cp->lo));
        count+=strip*top->Slice(a.top,a.bottom);
        top=top->Next();
    } while ((top) && (a.bottom>=top->lo));
    return count;
}

bool IsSet(Row* rp,Col* cp) const {
//Tests if a vixel is set,i.e. if the cells represented
//by the sub-area defined by 1 row and 1 column,are
//in the CellSelection.
    return rp->Vixel(cp->index);
}

public:
    CellSelection(void) {

        /* create an empty selection */

        CreateEmpty();
#ifdef EXTRA_FOR_TESTING
        testRow=0;
        testCol=0;
#endif
    }

```

```

}

~CellSelection(void) {

    /* free any allocated memory */

    FreeAll();
}

bool Clear() {

    /* make the selection empty */

    FreeAll();
    return CreateEmpty();
}

bool Add(Area area) {

    /* add the area of cells to this selection */

    if (!area.IsEmpty()) return SetArea(area);
    return true;
}

bool Remove(Area area) {

    /* remove the area of cells from this selection */

    if (!area.IsEmpty()) return ClearArea(area);
    return true;
}

bool Invert(Area area) {

    /* remove cells in the area that are also in this selection
       and add the area cells that are not in this
       selection */

    if (!area.IsEmpty()) return InvertArea(area);
    return true;
}

bool Add(const CellSelection & otherSelection) {

    /* add the otherSelection to this selection */

    Row* rp=otherSelection.row;
    do {
        Area a;
        a.top=rp->lo;a.bottom=rp->Limit();
        Col* cp=otherSelection.col;
        do {
            if (IsSet(rp,cp)) {
                a.left=cp->lo;a.right=cp->Limit();
                if (!SetArea(a)) return false;
            }
            cp=cp->Next();
        } while (cp);
        rp=rp->Next();
    } while (rp);
    return true;
}

bool Remove(const CellSelection & otherSelection) {

    /* remove the otherSelection from this selection */

    Row* rp=otherSelection.row;
    do {
        Area a;
        a.top=rp->lo;a.bottom=rp->Limit();
        Col* cp=otherSelection.col;
        do {
            if (IsSet(rp,cp)) {
                a.left=cp->lo;a.right=cp->Limit();
                if (!ClearArea(a)) return false;
            }
            cp=cp->Next();
        } while (cp);
        rp=rp->Next();
    } while (rp);
    return true;
}

bool Invert(const CellSelection & otherSelection) {

```



```

/* remove cells in the otherSelection that are also in this selection
and add the otherSelection cells that are not in this selection */

Row* rp=otherSelection.row;
do {
    Area a;
    a.top=rp->lo;a.bottom=rp->Limit();
    Col* cp=otherSelection.col;
    do {
        if (IsSet(rp,cp)) {
            a.left=cp->lo;a.right=cp->Limit();
            if (!InvertArea(a)) return false;
        }
        cp=cp->Next();
    } while (cp);
    rp=rp->Next();
} while (rp);
return true;
}
bool AllSelected(Area area) {

    /* return TRUE if all cells in the area are selected */

    if (area.IsEmpty()) return false;
    return AllSelected(area,true);
}
uint32 CountSelected(Area area) {

    /* count cells that are "on" */

    if (area.IsEmpty()) return 0;
    return CountSelectedP(area);
}
bool EqualSelected(const CellSelection & otherSelection) {

    /* return TRUE if otherSelection equals this selection */

    Row* rp=row;
    do {
        Area a;
        a.top=rp->lo;a.bottom=rp->Limit();
        Col* cp=col;
        do {
            a.left=cp->lo;a.right=cp->Limit();
            if (!otherSelection.AllSelected(a,IsSet(rp,cp)))
                return false;
            cp=cp->Next();
        } while (cp);
        rp=rp->Next();
    } while (rp);
    return true;
}

#if EXTRA_FOR_TESTING
const int Width() const {return width;}
const int Height() const {
    int h=0;
    Row* rp=row;
    while (rp) {h++;rp=rp->Next();}
    return h;
}
void ResetTestAreas() {
    testRow=row;
    testCol=col;
}
bool NextTestArea(Area* a) {
    while (testRow) {
        Row* rp=testRow;
        while (testCol) {
            Col* cp=testCol;
            testCol=cp->Next();
            if (IsSet(rp,cp)) {
                a->left=cp->lo;
                a->top=rp->lo;
                a->right=cp->Limit();
                a->bottom=rp->Limit();
                return true;
            }
        }
        testRow=rp->Next();
        testCol=col;
    }
}

```

```

        return false;
    }
#endif
};

#ifndef CELLS_DEFINITIONS_INCLUDED
#define CELLS_DEFINITIONS_INCLUDED

//We avoid Inlining of recursive methods
int Tree::Insert(Node* &p,Node* q,Node* prev) {

    // insert node q in (sub-)tree rooted in p
    int deltaH=0;
    if (p==0) {
        p=q;
        deltaH=1;
    }

    // also insert q in linear list, following prev
    if (prev) {
        p->nextNode=prev->nextNode;
        prev->nextNode=p;
    }
    else if (q->lo > p->lo) {
        if (Insert(p->rightNode,q,prev)) {
            p->bal++;
            if (p->bal==1) deltaH=1;
            else if (p->bal==2) {
                if (p->rightNode->bal==1)
                    RightRotate(p->rightNode);
                LeftRotate(p);
            }
        }
    }
    else if (q->lo < p->lo) {
        if (Insert(p->leftNode,q,prev)) {
            p->bal--;
            if (p->bal==1) deltaH=1;
            else if (p->bal==2) {
                if (p->leftNode->bal==1)
                    LeftRotate(p->leftNode);
                RightRotate(p);
            }
        }
    }
    return deltaH;
}

Node* Tree::Find(Node* p,int x) const {
    // find nearest p->lo <= x
    // never returns a NULL pointer
    if (p->lo<x) {
        if (p->rightNode) {
            Node* q=Find(p->rightNode,x);
            if (q->lo<=x) return q;
        }
    }
    if (p->lo>x) {
        if (p->leftNode) return Find(p->leftNode,x);
    }
    return p;
}
#endif

```

MT

**Want to know what products
are available for MacOS
development? Check out
Developer Depot™
<<http://www.devdepot.com>>**

How can you obtain the SUID for a class at runtime to determine compatibility? First, query the Virtual Machine for information about the class represented in the stream, using methods of the class `ObjectStreamClass`. Here is how we can get the SUID of the current version of the class named `MyClass`, as known to the Virtual Machine:

```
ObjectStreamClass myObject = ObjectStreamClass.lookup(
    Class.forName( "MyClass" ) );
long theSUID = myObject.getSerialVersionUID();
```

Now when we restore an `Externalizable` object, we can compare its SUID to the class SUID just obtained. If there is a mismatch, we should take appropriate action. This may involve telling the user that we cannot handle the restoration, or we may have to assign and use some default values.

If we are restoring a `Serializable` object, the runtime will check the SUID for us when it attempts to read values from the stream. If you override `readObject()`, you will want to compare the SUIDs there.

How do you determine what changes between class versions are acceptable? For an earlier version, which may contain fewer fields, trying to read a serialized object from a later version of the same class may cause problems. There is a tendency to add fields to a class as that class evolves, which means that the earlier version does not know about the newer fields. In contrast, since a newer version of a class may look for fields that are not present in the older version, it assigns default values to those fields.

This can be seen in the example code when we add a new field to the `MyVersionObject` class, but don't update the SUID. The new class can still read the older stream representation, even though no values exist in that stream for the new fields. It assigns 0 to the new int, and null to the new String, but doesn't throw any exceptions. If we then increment the SUID (from 1 to 2) to indicate that we do not consider older class versions compatible with this version, we throw an `InvalidClassException` when attempting to read a version 1 object from the stream.

The Sun documentation lists the various class format changes that can adversely affect the restoration of an object. A few of these include:

- Deleting a field, or changing it from non-static or non-transient to static or transient, respectively.
- Changing the position of classes in a hierarchy.
- Changing the data type of a primitive field.
- Changing the interface for a class from `Serializable` to `Externalizable` (or vice-versa).

On the other hand, not every change will have a negative effect. Here are some changes to class versions that do not have a detrimental effect on object behavior:

- Adding fields, which will result in default values (based on data type) being assigned to the new fields upon restoration.
- Adding classes will still allow an object of the added class to be created, since the class structure information is included in the stream. However, its fields will be set to the default values.
- Adding or removing the `writeObject()` or `readObject()` methods.
- Changing the access modifier (public, private, etc.) for a field, since it is still possible to assign a value to the field.
- Changing a field from static or transient to non-static or non-transient, respectively.

FORMAT OF A SERIALIZED OBJECT

The format for the default structure of a serialized object is similar, but not identical, to the structure of a class file. The Sun documentation describes in detail the format of the Object Serialization Stream. The example code writes files that may be opened with a text editor, so you can inspect the serialized objects.

EXAMPLE CODE

The following code illustrates the writing and reading of `Serializable` and `Externalizable` classes. `ObjectReaderWriter` is the primary application class. At runtime it displays a "Save As..." `FileDialog`, allowing you to specify an output file to receive the stream containing the serialized objects. (All the sample objects are written to the same file.) It then prompts for an input file from which to read a stream.

This arrangement of the sample code allows you to write out the serialized data to one file, make changes to the class format for one or more of the data classes, recompile and rerun, and attempt to read one of the older versions back in.

The class `MySerialObject` contains a reference to an instance of the class `MyInternalObject`, to demonstrate the saving of nested object references in the stream. `MySerialObject` also contains a field (of type int) that is marked transient, and upon restoration you will find that the default value 0 gets assigned to that variable.

The class `MyVersionObject` demonstrates the use of versioning with a programmer-specified SUID. You only need to change the SUID when you make changes to the class structure that render it incompatible with older versions of that same class, and whose serialized instances have previously been written to disk.

You can compile the .java (source) files using the `javac` (Java compiler) tool included in the MRJ SDK Tools folder, or using the Java compiler in CodeWarrior or Visual Cafe. You can then optionally create a .jar (Java archive) file containing the resulting .class (output) files.

The archive for this article includes the .java and .class files, and a .jar file containing the .class files. To run the program, drag either the file `ObjectReaderWriter.class` or `ObjectReaderWriter.jar` onto the JBindery application icon, which is located in the MRJ SDK JBindery folder. Once JBindery launches, it will display `ObjectReaderWriter` in the "class name" field. (This field specifies the name of the class to run at application startup; that class must contain a `main()` method.) Click OK to run the program.

Listing 1: ObjectReaderWriter.java

```

ObjectReaderWriter.java
The class that will read and write serialized and externalized objects.

import java.awt.*;
import java.io.*;

public class ObjectReaderWriter {
    String filePath;

    public static void main( String args[] ) {
        ObjectReaderWriter orw = new ObjectReaderWriter();
    }

    ObjectReaderWriter() {
        try {
            // Create instances of each data class to be serialized.
            MySerialObject serialObject = new MySerialObject();

            MyExternObject externObject = new MyExternObject();

            MyVersionObject versionObject = new MyVersionObject();

            // Allow the user to specify an output file.
            FileDialog fd = new FileDialog( new Frame(),
                "Save As...", FileDialog.SAVE );
            fd.show();
            filePath = new String( fd.getDirectory() + fd.getFile() );

            // Create a stream for writing.
            FileOutputStream fos = new FileOutputStream( filePath );

            // Next, create an object that can write to that file.
            ObjectOutputStream outStream =
                new ObjectOutputStream( fos );

            // Save each object.
            outStream.writeObject( serialObject );

            externObject.writeExternal( outStream );

            outStream.writeObject( versionObject );

            // Finally, we call the flush() method for our object, which forces the data to
            // get written to the stream:
            outStream.flush();

            // Allow the user to specify an input file.
            fd = new FileDialog( new Frame(), "Open...",
                FileDialog.LOAD );
            fd.show();
            filePath = new String( fd.getDirectory() + fd.getFile() );

            // Create a stream for reading.
            FileInputStream fis = new FileInputStream( filePath );

            // Next, create an object that can read from that file.
            ObjectInputStream inStream = new ObjectInputStream( fis );

            // Retrieve the Serializable object.
            serialObject = ( MySerialObject ) inStream.readObject();

```

```

            // Display what we retrieved:
            System.out.println( serialObject.getS() );
            System.out.println( "i = " + serialObject.getI() );
            serialObject.displayInternalObjectAttrs();

            // Retrieve the Externalizable object.
            externObject.readExternal( inStream );

            // Display what we retrieved:
            System.out.println( externObject.getS() );
            System.out.println( "i = " + externObject.getI() );

            // Retrieve the versioned object.
            versionObject = ( MyVersionObject )
                inStream.readObject();
            // Display what we retrieved:
            System.out.println( versionObject.getS() );
            System.out.println( "i = " + versionObject.getI() );

            // Display the SUID of the versioned class in the VM,
            // not necessarily the serialized object.
            ObjectOutputStream myObject = ObjectOutputStream.lookup(
                Class.forName( "MyVersionObject" ) );
            long theSUID = myObject.getSerialVersionUID();

            System.out.println
                ( "The SUID of class MyVersionObject = " + theSUID );
        }
        catch ( InvalidClassException e ) {
            System.out.println( "InvalidClassException..." );
        }
        catch ( ClassNotFoundException e ) {
            System.out.println( "ClassNotFoundException..." );
        }
        catch ( OptionalDataException e ) {
            System.out.println( "OptionalDataException..." );
        }
        catch ( FileNotFoundException e ) {
            System.out.println( "FileNotFoundException..." );
        }
        catch ( IOException e ) {
            System.out.println( "IOException..." );
        }
    }
}

```

Listing 2: MySerialObject.java

```

MySerialObject.java
The serializable data class.

import java.io.*;

public class MySerialObject implements Serializable {
    private transient int i;
    private String s;
    MyInternalObject mio;

    MySerialObject() {
        i = 64;
        s = new String( "Instance of MySerialObject..." );
        mio = new MyInternalObject();
    }

    public int getI() {
        return i;
    }

    public String getS() {
        return s;
    }

    public void displayInternalObjectAttrs() {
        System.out.println( mio.getS() );
        System.out.println( "i = " + mio.getI() );
    }
}

```


Listing 3: MyInternalObject.java

The nested data class.

```
MyInternalObject.java

import java.io.*;

public class MyInternalObject implements Serializable {
    private int i;
    private String s;

    MyInternalObject() {
        i = 128;
        s = new String( "Instance of MyInternalObject..." );
    }

    public int getI() {
        return i;
    }

    public String getS() {
        return s;
    }
}
```

Listing 4: MyExternObject.java

The externalizable data class.

```
MyExternObject.java

import java.io.*;

public class MyExternObject implements Externalizable {
    private int i;
    private String s;

    MyExternObject() {
        i = 256;
        s = new String( "Instance of MyExternObject..." );
    }

    public int getI() {
        return i;
    }

    public String getS() {
        return s;
    }

    public void writeExternal( ObjectOutputStream out ) throws
        IOException {
        out.writeInt( this.i );
        out.writeObject( this.s );
    }

    public void readExternal( ObjectInputStream in ) throws
        IOException, ClassNotFoundException {
        this.i = in.readInt();
        this.s = ( String )in.readObject();
    }
}
```

Listing 5: MyVersionObject.java

The versioned data class.

```
MyVersionObject.java

import java.io.*;

public class MyVersionObject implements Serializable {
    static final long serialVersionUID = 1L;
    private int i;
    private String s;
}
```

```
// Uncomment the next two lines to verify that default values will be substituted if
// the value is not present in the stream at deserialization time.
// private int i2 = -1; private String s2 = "This is the new String field";
```

```
MyVersionObject() {
    i = 512;
    s = new String( "Instance of MyVersionObject..." );
}

public int getI() {
    return i;
}

public String getS() {
    return s;
}
}
```

CONCLUSION

Adding object persistence to Java applications using serialization is easy. Serialization allows you to save the current state of an object to a container, typically a file. At some later time, you can retrieve the saved data values and create an equivalent object. Depending on which interface you implement, you can choose to have the object and all its referenced objects saved and restored automatically, or you can specify which fields should be saved and restored. Java also provides several ways of protecting sensitive data in a serialized object, so objects loaded from a serialized representation should prove no less secure than those classes loaded at application startup. Versioning provides a measure of the backward compatibility of class versions. The code needed to add serialization to your application is simple and flexible.

REFERENCES

Developing Java Beans, Robert Englander, O'Reilly & Associates, Inc., 1997.

URLs

- <http://www.javasoft.com/products/jdk/1.1/docs/guide/serialization/index.html>.
- <http://www.apple.com/macos/java/>.



**Interested in advertising in
*MacTech Magazine?***

**Contact our
ad sales department at
<mailto:adsales@mactech.com>**



by Steve Sisak

It is easier to quit an app than to close a window because the close window event requires a direct object and a position descriptor to indicate which window. Quitting an app simply requires creating a quit event and sending it.

The following code takes a valid target desc (which I'm assuming you already know how to create) and sends a close first window event to the app specified in that target desc.

The code is modified from similar code I used a while back. I may have made some errors in removing stuff or changing identifiers, but it should give you the right idea. Feel free to e-mail any questions about it.

```
void CloseFrontWindowOfApp( AEAddressDesc * theTargetDescPtr )
{
    AEDesc      frontWindowObjSpecifier;
    AppleEvent  event;
    AppleEvent  reply;
    OSStatus    err = noErr;

//initialize descriptors
    event.descriptorType = typeNull;
    reply.descriptorType = typeNull;
    frontWindowObjSpecifier.descriptorType = typeNull;

    event.dataHandle = NULL;
    reply.dataHandle = NULL;
    frontWindowObjSpecifier.dataHandle = NULL;

//create object spec for front window
    err = MyCreateFrontWindowObjSpec( &frontWindowObjSpecifier );
    if ( err )
        goto CLEANUP_SetBounds;

//create apple event- not sure if close event is in kAECoreSuite?
    err = AECreatAppleEvent( kAECoreSuite, kAEClose,
                             theTargetDescPtr, kAutoGenerateReturnID,
                             kAnyTransactionID, &event );
    if ( err )
        goto CLEANUP_SetBounds;

//insert direct object into ae
    err = AEPutParamDesc( &event, keyDirectObject,
                          &frontWindowObjSpecifier );
    if ( err )
        goto CLEANUP_SetBounds;

//send ae
    err = AESend( &event, &reply, kAENoReply +
                  kAENeverInteract +
                  kAECanSwitchLayer,
                  kAENormalPriority, kAEDefaultTimeout,
                  kNoIdleProc, kNoFilterProc );
}
```

```
//dispose descriptors
CLEANUP_SetBounds::
    AEDisposeDesc ( &event );
    AEDisposeDesc ( &reply );
    AEDisposeDesc ( &frontWindowObjSpecifier );

} //end function

OSStatus MyCreateFrontWindowObjSpec( AEDesc *frontWindowObjSpec )
{
    AEDesc      emptySpecifier;
    AEDesc      positionDesc;
    long        thePosition;
    DescType    propertyType;
    OSStatus    err;

//initialize descriptors
    emptySpecifier.descriptorType = typeNull;
    positionDesc.descriptorType = typeNull;
    propertyDesc.descriptorType = typeNull;

    emptySpecifier.dataHandle = NULL;
    positionDesc.dataHandle = NULL;
    propertyDesc.dataHandle = NULL;

//create objspecifier for window 1 of null container
    thePosition = 1;
    err = AECreatDesc( typeLongInteger, (Ptr)(&thePosition),
                      sizeof( thePosition ), &positionDesc );
    if ( err )
        goto CLEANUP_MyCreateBoundsObjSpec;

    err = CreateObjSpecifier( cWindow,
                              &emptySpecifier,
                              formAbsolutePosition,
                              &positionDesc,
                              false,
                              frontWindowObjSpec );

    if ( err )
        goto CLEANUP_MyCreateBoundsObjSpec;

//dispose descriptors
    CLEANUP_MyCreateBoundsObjSpec::
        AEDisposeDesc ( &windowObjSpecifier );
        AEDisposeDesc ( &positionDesc );

    return err;
}
```

David T. Pierson
dtp@pluto.njcc.com

MT

Send us your tips or we'll install *EvenBetterBusError* on your machine! On the other hand, we might just pay you \$25 for each tip we use, or \$50 for Tip of the Month. You can take your award in goods, subscriptions or US\$. Make sure any code compiles, and send tips (and where to mail your winnings) to our **Tips e-mail address** at tips@mactech.com. (See page 2 for our other addresses.)

by Jeff Clites <online@mactech.com>

Whoever first said, "It's a small world," must not have been shopping for a database. There is a lot out there, probably because the idea of a database is one of the most fundamental and powerful concepts in computing. After all, what do computers do if not manipulate, store, and retrieve data? At its simplest, a database can be just a file system. On another level, it can be a mechanism for providing object persistence. And on a larger scale, it can be a repository of information to be accessed remotely from thousands of locations and multiple platforms. To help make sense of it all, we're going to take a quick tour of this vast landscape. Since several of the big tourist spots are reviewed elsewhere in this issue, we're going to visit a few of the more intimate, out of the way spots that might otherwise be missed.

DON'T FORGET TO WRITE HOME

As with all trips, it can be surprising how many things there are to see right at home — and in this case, home is Apple. Developer World has an extensive list of products available for the Macintosh for client/server and database development. It's over a year old, but still provides a wealth of information about just what is out there, and it gives brief explanations of some of the terminology involved along the way. There is also the MacODBC SDK, which uses a shared library to provide a generalized API for interacting with databases.

While you're in the Apple domain, stop by the AppleScript site, where there is a tutorial on database publishing. They use FileMaker Pro and QuarkXPress to demonstrate how AppleScript can be used to bring the services of a database to another scriptable application. It's a good place to start convincing yourself that AppleScript really *can* be used by serious developers.

Here's one of the best kept secrets of the Mac OS: Apple has cautioned developers that the resource manager is not a database. But, then they "thought different" and provided the Dictionary Manager, which is. It's designed to provide a reusable format for dictionaries used by input methods for 2-byte script systems, but it can also be used for other lightweight database tasks. If you can handle its restrictions (keys at most 129 bytes, records at most 4096 bytes, and a total database of less than 16MB) or are just getting your feet wet with database programming, then it can provide a simple, free, and always available alternative to third-party packages. It's documented in Inside Macintosh: Text.

Client/Server and Database Development Tools for the Macintosh

<<http://devtools.apple.com/general/Guide2MacTools/csdatabase.html>>

Apple Software Development Kits

<<http://devworld.apple.com/ngs/pp/adpub/docs/dev/sdk.html>>

Database Publishing Lessons

<http://applescript.apple.com/applescript_overview/dbase_lessons/DBASE_LESSON.00.HTML>

The Dictionary Manager

<<http://gemma.apple.com/dev/techsupport/insidemac/Text/Text-483.html>>

LEARN THE LANGUAGE

The database market comes with its own set of lingo and acronyms. If you're a little befuddled by terms such as SQL, ODBC, DBMS, and 4GL, then the Free On-line Dictionary of Computing can help.

But when you want to start getting down to serious business, it's time to learn the to talk directly to your database — like a native. Fortunately, there is a standard language, which is used by many (but certainly not all) commercial databases. It's SQL, the Structured Query Language, and it's spoken by the big boys, Oracle, Sybase, and Informix, as well as by others. For the brave, there is an online tutorial and an FAQ, but the most useful resource may be the "Ask the SQL Pro" site, which can help answer your questions as you learn the language.

Free On-line Dictionary of Computing

<<http://wfn-shop.princeton.edu/cgi-bin/foldoc>>

SQL Tutorial

<<http://w3.one.net/~jhoffman/sqltut.htm>>

SQL FAQ

<http://epoch.CS.Berkeley.EDU:8000/sequoia/dba/montage/FAQ/SQL_TOC.html>

Ask the SQL Pro

<<http://www.inquiry.com/techtips/thesqlpro/>>

PLANNING YOUR NEXT TRIP

There are two things certain to be in the future of any Mac programmer: Rhapsody, and Java. You can't hide from them forever. Fortunately, both come ready-made to interact with database servers. Rhapsody has the Enterprise Objects Framework, which allows your applications to interface with relational databases, and which most notably allows WebObjects applications to interact with a database in an object-oriented manner. Rhapsody is also slated to ship with a free version of the OpenBase SQL database engine.

For Java developers, life is good, because there is an SQL database interface which forms a standard part of Java, and which is included in the JDK 1.1. It's the JDBC API, and you can find out all about it on Sun's Java site.

Documentation for Internet/Enterprise Programmers

<<http://gemma.apple.com/techinfo/techdocs/enterprise/enterprise.html>>

WebObjects 3.5 Documentation

<<http://gemma.apple.com/techinfo/techdocs/enterprise/WebObjects/WebObjectsTOC.html>>

Creating a WebObjects Database Application

<<http://gemma.apple.com/techinfo/techdocs/enterprise/WebObjects/GettingStarted/Movies/MoviesTOC.html>>

OpenBase International, Ltd.

<<http://www.openbase.com/>>

The JDBC(tm) Database Access API

<<http://java.sun.com/products/jdbc/index.html>>

These and other links are available from the MacTech Online web pages at <<http://www.mactech.com/online/>>. 

BestBusError

Mr. BusError causes a BusError if you dereference NIL. EvenBetterBusError and DavesBusError improved on this feature, providing a BusError if you write to a NIL pointer or execute code at NIL. The BestBusError tops them all... it provides a BusError no matter WHAT you do!

*Ask us about the BestBusError Deinstaller Kit! 1-800-555-3665
<<http://www.MacTech.com/BestBusError/>>*

*"It has doubled the number of bug reports we receive!"
- Poynt T. Hayrd-Boss*

*"Please don't crash, please don't crash..."
- Nick DeMello (Beta Tester)*

*"One of the most memorable products we have
ever reviewed."
- MacTech Magazine*

*"I love the new random timer feature!"
- April F. Ool (Quality Control Manager)*

*"It does suck."
- Rich Siegel (Author of BBEdit)*

For a limited time only:

\$99

Offer expires April 2nd 1998

Coming In April '99 Version 2.0 with Quit Function!

Windows version unnecessary!

Great as a gift!

Unclear Concept Software Inc.

Makers of "the System Uninstaller" and "DiskDeleter"

**Not Nominated for
ANYTHING!**



by Jessica Courtney

STONE DESIGN AND CAFFEINE SOFTWARE DELIVER THE ENHANCED LICENSER KIT

Long time NeXTSTEP/OPENSTEP/Rhapsody developers, Stone Design Corp, makers of Create, and Caffeine Software, makers of TIFFany, announced today that their acclaimed Licenser Kit has been updated and improved, and future upgrades of the kit are free, at least until the year 2001.

The Licenser Kit allows developers to quickly add the level of copy-protection security they require, including the following, combinable options for generating restricted usage:

- Tie license to user name.
- Tie to host ID (now including Windows 95/NT!)
- Create floating network licenses.
- Allow as many users as desired.
- Have unlimited number of licenses on a network.
- Add licenses on the fly (cumulative licensing).
- Automatically unlicense if application is moved or copied.
- Generate expiring licenses for "full strength" demos.
- Automatically track the licenses currently in use on the network.
- Fully customizable through subclassing and interface files.
- Only you know the encryption code for your apps.
- Full source for License Generation app included.
- Royalty free.
- Free upgrades at least until 1/1/2001.

The Licenser Kit has been in development and testing for over five years by Caffeine Software and Stone Design, and is the choice software protection solution for all the top Rhapsody developers, including Anderson Financial Systems Inc., P&L Systems Ltd., OneStep Solutions Plc., Jenike & Johanson, Inc., Robert Vasvari, and other not yet publicly disclosed Rhapsody and YellowBox developers.

Added to this release is a network license monitoring system that permits users to see who is using the same application, which is very helpful when all of the licenses are already in use. The people using the licenses and the machines they are using them on are automatically listed in the Licenser Kit panel.

Another new feature is the ability for a software house to have licenses generated for one application that can be used in another application, but not vice-versa. This comes in handy when developers offer a "pro" product along with a "companion" application, for example.

The Licenser Kit is available now for Rhapsody Developer Release, Apple YellowBox for Windows, OPENSTEP 4.x, OPENSTEP Enterprise for Windows and NT, and NeXTSTEP. Unlike competing products, The Licenser Kit supports network

floating licenses, where a license can be installed on the network and shared by all users, which gives the developers better copy protection control and is extremely end user friendly. The floating licenses are well suited for site licensing. Moreover, the Licenser Kit enables users to purchase an initial number of floating licenses, and later increase this number using the Kit's cumulative license strategy. Of course, expiring (timebomb) licenses can be generated on the fly by the developer for fully functional demonstrations.

<http://www.stone.com/Licenser/>

ACI PRODUCT LINE v6.0.5

Less than a year after the release of its major product upgrade, 4th Dimension version 6.0, ACI has engineered a significantly enhanced version for the entire ACI Product Line — version 6.0.5.

The benefits provided by this maintenance release include improved functionalities and an extremely stable and robust design environment. ACI Product Line version 6.0.5 is proof of ACI's commitment to providing quality development tools to its 7,000-strong developer community. Version 6.0.5 also includes some new features, such as new management of picture field compression, and a new procedure for converting 4D version 3 applications to 4D version 6.

The success of 4th Dimension version 6.0, released in early 1997, can be measured by the large proportion of professional developers who have already upgraded to this new version to build powerful cross-platform applications. Among the highly acclaimed features of 4th Dimension and 4D Server v6 are a built-in Web server, a revamped interface, triggers, stored procedures, a visual, interactive debugger, and an object-based forms editor.

<http://www.acius.com>

FAIRCOM RELEASES NEW VERSION OF C-TREE PLUS, V6.7A, FOR MACINTOSH WITH CONDITIONAL INDEX SUPPORT, TRANSACTION HISTORY, VARIABLE LENGTH API

FairCom Corporation announced the latest release of c-tree Plus v6.7A file handler and the FairCom Server for the Macintosh, including support for System 7.X and Mac OS 8.X for both Native PowerPC and 68K. This new release of FairCom's C ISAM database API offers an enhanced variable length API resulting in easier and faster variable length record operations and conditional index support, which provides a means to define complex expressions and use them to filter the entries within an index. Other major features of this release include a transaction history function, which makes it possible to access the audit logs of transaction controlled files.

Developer Central™

"The Place for Macintosh Developers"

Sponsored by



And



**Worldwide Developers Conference
(WWDC)**

MAY 11 - 15, 1998

San Jose Convention Center

At Developer Central, YOU CAN...

- Talk directly to representatives from Apple and third-party vendors
- Demo developer products and tools
- Get the best deals on tools, training, and developer resource materials

So, no matter if you're developing a departmental system, developing plug-ins for the Internet, or planning the next great Mac OS application or multimedia title, you'll find what you need at **Developer Central**.

<http://www.devcentral.com>

Apple, the Apple logo, Macintosh, Mac, and the Mac OS logo are registered trademarks of Apple Computer, Inc. Developer Central, Virtual Developer Central are trademarks of Apple Computer, Inc. MacTech is a registered trademark of Xplain Corporation.

The new release of c-tree Plus also provides conditional index support, which allows indices to be defined at the data content level rather than at just the field level. Dynamic index (select/omit) filtering can be performed on c-tree Plus index files by using a powerful expression parser/analyzer. Complex expressions are defined and evaluated at run time to control index entries, resulting in smaller, highly efficient indices groomed to a developer's precise needs. The conditional index support is available in all seven c-tree Plus operational modes.

The transaction history function allows for the accessing of the audit logs of transaction controlled files. This enables the programmer to scrutinize the changes performed on each unit of information at a very detailed level. One such use of this technology has been to track the changes made to a manufactured device as it proceeds through an automated fabrication factory. The exact individual history of each manufactured device can be determined from the transaction log files.

Additional features of version 6.7 of c-tree Plus include: ISAM access enhancements, batch performance, system configuration function and transaction processing enhancements; new extended performance monitor; and file path control.

FairCom created Access Manager for Digital Research in 1979 and released its first true application development tool, the original c-tree file handler, that same year. Today c-tree Plus is running in over 100 CPU/OS environments in more than 98 countries worldwide. FairCom Macintosh implementations are utilized by such noted organizations as: McGraw-Hill, Follett Software Company, Great Plains Software and Phone Directories. Corporations like Computer Associates and Sharp Corporation have embedded FairCom's c-tree Plus within vertical market products and licensed its Server technology for OEM uses
<http://www.faircom.com>

QuickCRC 1.1

OBJECT-ORIENTED SOFTWARE DESIGN TOOL

Excel Software is shipping QuickCRC 1.1 for Macintosh and Windows 95/NT. QuickCRC is a design tool for discovering objects and related information for an object-oriented software development project. It automates the CRC card concept of identifying classes, responsibilities and collaborations between objects. Design scenarios involving a group of communicating objects can be identified and simulated. New features in this release include a contents view, inheritance graph and full text import and export of design information.

QuickCRC uses a diagram workspace for creating card and scenario objects. A card represents the properties of a class including its name, description, superclasses, subclasses, attributes, responsibilities and collaborating objects. A scenario represents a design mechanism defined as a series of steps involving communicating objects. Scenarios can reference cards or other scenarios. Cards and scenarios can also reference external agents defined by the designer to identify system and user interfaces. As information is entered or changed for a card or scenario object, it is instantly synchronized throughout the model.

Separate diagrams partition a model into subject areas. The contents view allows a designer to navigate between diagrams shown as folder icons that can be opened or closed. From the contents view, a card or scenario can be dragged between diagrams or double-clicked to edit its properties. This makes it easy to locate and modify information as the model grows.

The inheritance graph concisely illustrates the class inheritance structure of the model. It gets generated from information on the CRC cards regardless of how the cards are physically arranged or partitioned between diagrams. Several inheritance graphs can be active for different root classes in the CRC model to focus attention on different parts of a large design. The properties of a card can be viewed or edited by double-clicking its name on the inheritance graph.

The List Diagrams command generates a text representation of the information in a CRC model in a syntactically simple format. This information can be transferred to other applications or used to generate a new CRC model with the Import Diagrams command. Design models can be exported or imported to Excel Software's MacA&D and WinA&D software engineering tools for detailed design or code generation. Likewise, Excel Software's reengineering tools can be used to generate QuickCRC design models from existing code.

<http://www.excelsoftware.com>

AAA+ SOFTWARE SHIPS JOY RELEASE 1.08 FOR OPENSTEP, RHAPSODY AND WINDOWS

AAA+ Software announces release 1.08 of Joy Explorer and Joy Developer, the company's developer productivity and rapid application development tools. Joy applications run on Windows 95/NT, Apple Rhapsody, and OPENSTEP/MachOS. No porting or compiling is required.

Release 1.08 offers enhancements including:

- Joy graphical application inspector integrated into Joy command window.
- Joy Developer's "Save Nib as App" now creates stand-alone applications for use on any deployment platform.
- Linking Joy into non-Joy applications made easier

Joy Explorer is the ideal tool for exploring the Yellow Box frameworks. Developers can create small utilities by using just Rhapsody's Joy-enhanced Interface Builder application. No other development tools are required. These utilities will run in Interface Builder's Test Interface mode.

Joy Developer provides full functionality for developing sophisticated applications. There is no need to re-compile, re-link and re-establish test scenarios to explore new ideas. Using the "Save Nib as App" feature, users can turn graphical user interfaces created in Interface Builder into standalone applications that run on any OPENSTEP or Yellow Box platform.

<http://www.aaa-plus.com/joy/>

<http://www.aaa-plus.com/joy/download.html>



We figured
a lowest price
guarantee would be cool.
Then, someone suggested a
30-day satisfaction guarantee
...and we thought we'd do that
too. Then, someone else called
every vendor they could think of,
and stocked hundreds of tools,
books, accessories and everything
else a developer/programmer would
ever want. Then, we set up toll-free
phones, an awesome Web site, fax
lines, and e-mail accounts so our
customers could order as conveniently
as possible. Finally, some dude
goes: "hey, let's just give
away products for FREE!"
Yeah right.



E-mail: orders@devdepot.com • Web site: <http://www.devdepot.com>
Phone: 800-MACDEV-1 - 805-494-9797 (outside the U.S. & Canada) • Fax: 805-494-9798

SERVICES

The Law Office of Bradley M. Sniderman

California Lawyer focusing on Intellectual Property, Corporate, Commercial and Contract law, as well as Wills and Trusts. If you are looking to protect your software with Copyright or Trademark protection, or if you need help establishing or maintaining your business, please give me a call or an e-mail. Reasonable fees.

(310) 553-4054

Brad@Sniderman.com

MACINTOSH PROGRAMMER

Macintosh programmer needed for challenging position to assist in development of cross-platform experiment-authoring package for psychologist. Software allows users to author and run real-time cognitive experiments using scripting language or graphic interface. Help us to create a first-class Macintosh product. New positions immediately available at small Pittsburgh area software firm. Send cover letter, resume and salary requirements to Human Resources, Psychology Software Tools, 2014 Monongahela Avenue, Pittsburgh, PA 15218 or e-mail to info@pstnet.com. See www.pstnet.com for more information.

<http://www.scientific.com>

Professional Software Developers

Looking for career opportunities?

Check out our website!

Nationwide Service

Employment Assistance

Resume Help

Marketability Assessment

Never a fee

Scientific Placement, Inc.

800-231-5920 800-757-9003 (Fax)

das@scientific.com

Wanna see your product *move*?



Get it in Developer Depot!

The fastest, cost effective way to get product off your shelves.

For information: • Voice: 805/494-9797 • Fax: 805/494-9798 • E-mail: vendor_relations@devdepot.com

**If you want to be in the know, then you
need every article published in the first
12 years of MacTech® Magazine and
Apple's develop™ issues 1-29!**

...in THINK Reference™ format!

**So hurry, pick up the phone, fire up
the e-mail, launch that fax machine,
or simply drop by our web site and
order yourself this new release of
the MacTech® CD-ROM.**

- Almost 1600 articles from all 139 issues of MacTech Magazine (1984-1996)
- Includes Apple develop issues 1-29
- Improved hypertext, improved indices, and a new THINK Reference Viewer — for lightning quick access!
- New hyperlinks between articles
- 100+ MB of source code — use them in your applications, with no royalties!
- Full version of THINK Reference — the original online guide to Inside Macintosh, Vols. I-VI
- 80MB of FrameWorks/SFA archives and the most complete set of FrameWorks archives known
- Sprocket™! MacTech's tiny framework that compiles quickly and supports System 7.5 features
- The best threads from the Mac programmer newsgroups plus thousands of notes, tips, snippets, and gotchas
- Popular tools that Mac programmers use to increase their productivity and much more!



Developer DEPOT™

Web Site: <http://www.devdepot.com> • E-mail: orders@devdepot.com

Phone: 800-MACDEV-1 • Outside the U.S. & Canada: 805-494-9797 • Fax: 805-494-9798

LIST OF ADVERTISERS

Aladdin Knowledge Systems Ltd.	5
APS Technologies	71
Bare Bones Software, Inc.	13
Blue World	7
Creative Solutions	37
Developer Depot	75
Earthlink	11
Mac Hack	9
MacTech CD ROM	77
MacTech Web Site	32
Mathemaesthetics, Inc.	17
Metrowerks	BC
Mitsubishi	1
Neologic	15
Onyx Technology, Inc.	31
OpenBase International	29
Optic Storage	45
PrimeTime Freeware	43
Psychology Software Tools	76
Quadrivio Corporation	35
Seapine Software, Inc.	23
Scientific Placement	76
SLab	41
SNAP	45
Snowbound Software Corporation	21
Sophisticated Circuits	25
Stone Design	19
StoneTablet Publishing	49
Symantec	IFC
Water's Edge Software	39
WWDC	73

LIST OF PRODUCTS

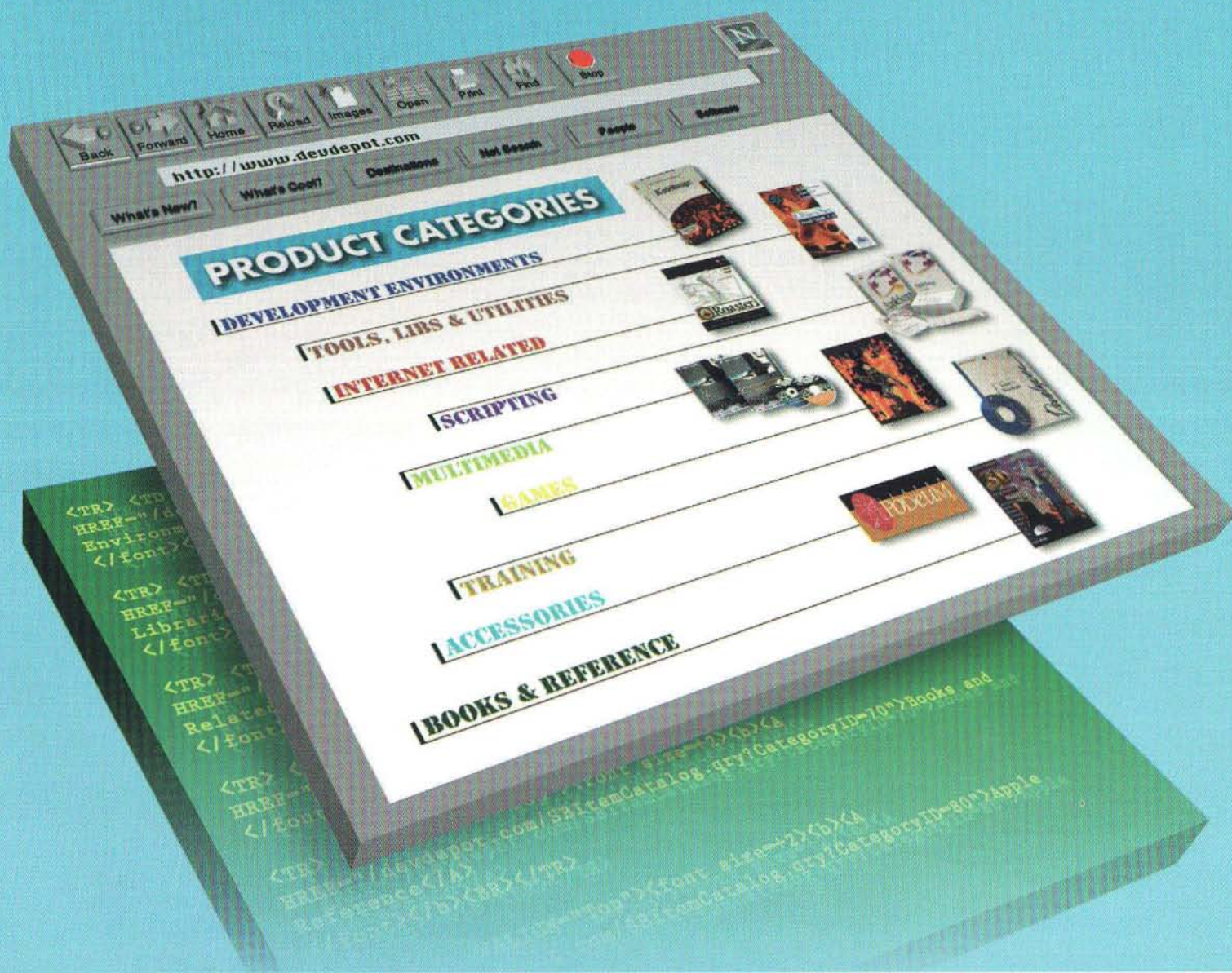
BBEdit • Bare Bones Software, Inc.	13
Classified • Scientific Placement, Inc.	76
Classified • Sniderman, Bradley M.	76
Classified • Psychology Software Tools	76
Clone It • Optic Storage	45
CodeWarrior™ • Metrowerks	BC
Developer Tools • Developer Depot	75
Diamond Plus Series/Diamond Pro Series • Mitsubishi	1
dtF/SQL • SLab	41
General Edit • Quadrivio Corp.	35
Internet Access • Earthlink	11
Lasso 2.5 • Blue World	7
Licenser Kit • Stone Design	19
MACHASP • Aladdin Knowledge Systems Ltd.	5
MK Linux/MacPerl • PrimeTime Freeware	43
NeoAccess • Neologic	15
OpenBase • OpenBase International	29
PowerKey Pro • Sophisticated Circuits	25
PrimeBase • SNAP	45
RasterMaster 6.0 • Snowbound Software Corporation	21
Resourcerer® 1.2 • Mathemaesthetics, Inc.	17
Serial/Parallel Expansion Cards • Creative Solutions	37
Spotlight™ • Onyx Technology, Inc.	31
StoneTable • StoneTablet Publishing	49
TestTrack™ • Seapine Software, Inc.	23
Tools Plus™ • Water's Edge Software	39
Trade Show • Mac Hack	9
Visual Cafe™ • Symantec	IFC
World Wide Developers Conference • WWDC	73

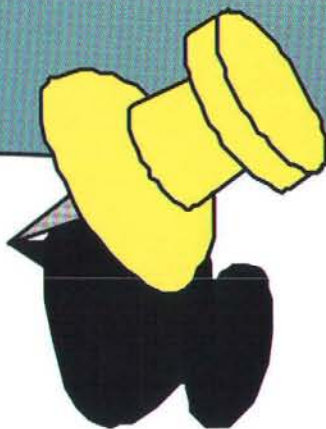
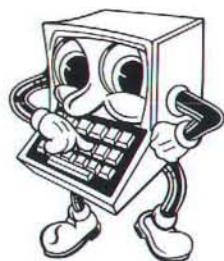
The index on this page is provided as a service to our readers. The publisher does not assume any liability for errors or omissions.

Developer DEPOT™

Developer Depot PO Box 5200 Westlake Village CA 91359-5200
800/MACDEV-1 • Outside the U.S. & Canada: 805/494-9797 • Fax: 805/494-9798
E-mail: orders@devdepot.com • Web Site: <http://www.devdepot.com>

- ✓ World renowned customer service
- ✓ Order by phone, E-mail, fax or through our continually updated Web site
- ✓ Hundreds of developer products
- ✓ Satisfaction and lowest price guaranteed





Developer To Do List:

- ☒ **Prioritize project list & generate schedule**
- ☐ **Pick up new MacTech CD-ROM**
- ☐ **Check out all the Dev Depot specials**
- ☐ **Tell the gang about Dev Depot deals**
- ☐ **Stock up on the latest books & utilities**
- ☐ **Enter Dev Depot info into contact file**

Developer Depot
PO Box 5200
Westlake Village, CA 91359-5200

Voice: 800/MACDEV-1
Outside US/Canada: 805/494-9797
Fax: 805/494-9798
E-mail: orders@devdepot.com
<http://www.devdepot.com>

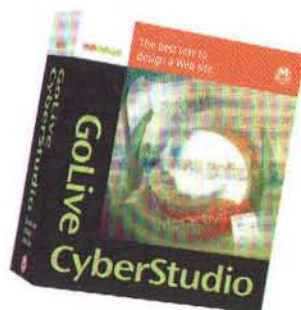
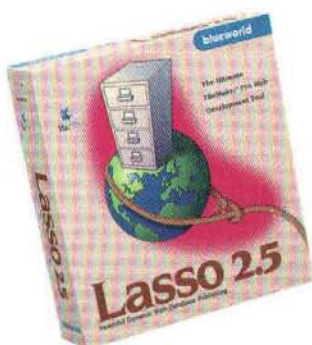
Developer **DEPOT**™

ANNOUNCEMENT

Developer DEPOT™

Spring Sale

April 6 - April 12



KAIDAN

Digital Imaging Technology

02	DEVELOPMENT ENVIRONMENTS
06	TOOLS, LIBS & UTILITIES
12	INTERNET RELATED
14	SCRIPTING
15	MULTIMEDIA
19	GAMES
20	ACCESSORIES
21	BOOKS & REFERENCE
31	INDEX

TABLE OF CONTENTS



Order Toll-free
800-MACDEV-1
(800-622-3381)

Developer Depot 30 day Money Back, Price and Satisfaction Guarantee

Developer Depot products are sold with a 30 Day money back guarantee on user satisfaction, lower prices and against defects. If, for any reason, you are not satisfied or find the same product at a lower price within 30 days, please call Customer Service at 800-MACDEV-1 and request a Return Merchandise Authorization (RMA) number to get a full refund or the difference in price (where applicable). You must return undamaged product at your expense, including all its original packaging, documentation and the blank warranty

card if applicable. Developer Depot will replace defective product upon receipt of the defective merchandise. Please remember to back up your data before installation of any new hardware, software, or peripherals; we cannot be responsible for any lost data. Policies, item availability, and prices are subject to change without notice. The price in effect when we receive your order will be the price that you are charged. We are not responsible for any typographical errors in this or any other catalog, nor for any misstatements from any vendor. Purchase orders are not accepted without prior approval. Call for more information.

Stuff our lawyer made us write.
Developer Depot makes no other warranties. All other warranties, either expressed or implied, including the implied warranty of merchantability and fitness for a particular purpose are disclaimed. Developer Depot shall not be liable for any direct, special, incidental or consequential damages including lost profits, from any delay in delivery, or for any personal injury arising from the use of any product sold through Developer Depot. The limit of direct damages, if any, shall not exceed the purchase price of the product. © 1997 Xplain Corporation. All rights reserved. Any unauthorized duplication is in violation of federal laws. Developer Depot is a registered trademark of Xplain Corporation. All product names in this catalog are the trademarks of their respective holders.

© 1997 Xplain Corporation. All rights reserved. Any unauthorized duplication is in violation of federal laws. Developer Depot is a trademark of Xplain Corporation. All product names in this catalogue are trademarks or registered trademarks of their respective holders

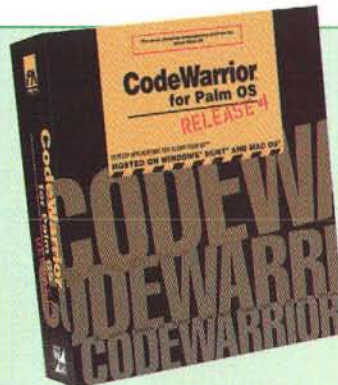


CodeWarrior for PalmPilot Release 4

by Metrowerks

CodeWarrior for PalmPilot is the first complete set of tools which enables you to develop for the U.S. Robotics PalmPilot connected organizer, from the comfort of your PC or Macintosh computer. All the tools you need are included: the award-winning CodeWarrior IDE, compiler, linker, source-level debugger, GUI builder, and other tools, plus online documentation and reference materials. Includes one free product update and free technical support with registration.

(SCWPALM) Our Price **\$369**



Check out our Web site!

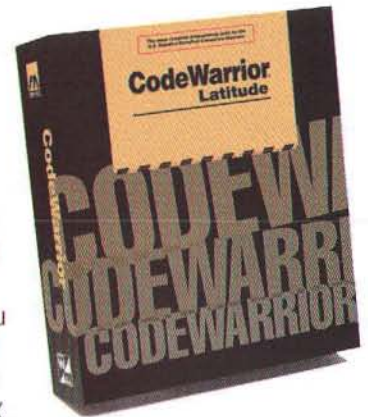
• Full product descriptions • Hundreds of more products
<http://www.devdepot.com>

CodeWarrior Latitude

by Metrowerks

Don't throw away the investment you have made in your Mac OS application! With the new DR2 release of CodeWarrior Latitude, you can now port your Mac OS application to Rhapsody, as well as Silicon Graphics IRIX and Sun Solaris. At the heart of Latitude is a set of shared libraries which performs the functions of the Macintosh API. You recompile your Mac OS source code, linking it with the Latitude libraries to produce a native application. As the Rhapsody API evolves, so will Latitude. Registered users of CodeWarrior Latitude will receive all developer releases, the first full release, plus one additional product update, to ensure that you have access to the most up-to-date Rhapsody porting tools available.

(SCWLAT) Our Price **\$399**

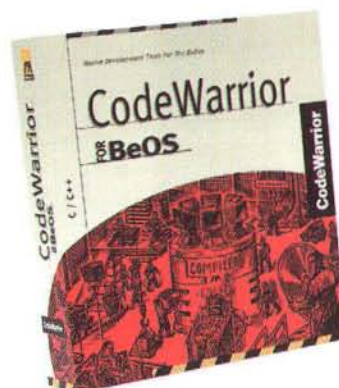


CodeWarrior for BeOS 3

by Metrowerks

- Start programming for the new, innovative Be Operating System (BeOS) with complete set of Codewarrior tools
- BeOS-native Integrated Development Environment (IDE) with all the familiar CodeWarrior features at your fingertips
- A BeOS PowerPC compiler and linker, an editor w/syntax color and styling, and a source-level debugger
- BeOS header and libraries, complete documentation, useful C++ classes, and sample code
- Now includes Java support
- The BeOS Preview Release allows you to run and program for the BeOS on a 603 or 604 PCI based PowerMac

(SCWFB) Our Price **\$299**

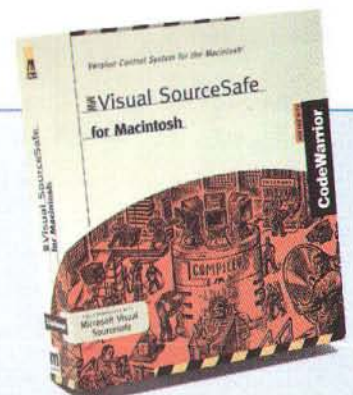


MW Visual SourceSafe Release 5

by Metrowerks

- Source code control system, plug-in to the CodeWarrior IDE or stand-alone Client application
- Compatible with Microsoft Visual SourceSafe version 5.0
- Cross-platform support (Mac, Windows, UNIX)
- Configuration management in excess of 4 billion files
- Over 8,000 files and sub-projects in a single sub-project
- Registered users receive one year of free technical support and one free update

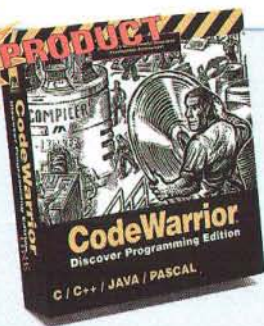
(SMWVSS) Our Price **\$499**



CodeWarrior Discover Programming Edition

by Metrowerks

- Learn to program in C/C++/Java/Pascal
 - Full-featured programming product—not a "lite" edition
 - Online books and tutorials
 - Hosted on either Mac OS or Windows 95/NT
- CodeWarrior Discover Programming Edition offers an unbeatable combination of full-featured programming tools, online books and instructional materials, all at a great price. Whether you want to learn the basics of programming or add a new language to your skill set, Discover Programming puts the information and tools you need at your fingertips: the award-winning, easy-to-use CodeWarrior Integrated Development Environment (IDE), four of the most popular programming languages, online books, online tutorials, sample source code and free technical support (with your registration). Discover what you can learn with CodeWarrior, buy your copy today!
- (SDPED) Our Price **\$79**

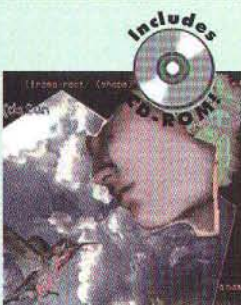


Pro Fortran

by Absoft Corporation

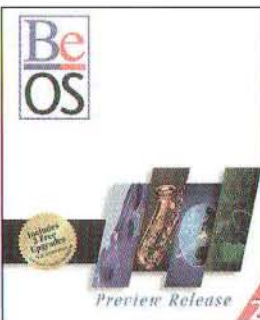
Absoft Pro Fortran combines native F90, VAX compatible F77, and C/C++ compilers into a single, easy to use environment. All compilers are link compatible and operate through a common interface.

- Graphical debugger, browsers, array display, performance profiler, linker, MRWE application mainframe
 - MIG graphics library, Absoft Create Make, several utilities, the latest version of MPW and illustrated documentation
 - Whole array operations, modules, interface blocks, and user-defined types or data structures
 - Dynamic memory allocation and new control constructs
 - F90 is link compatible with Absoft F77, C++, MrC and CodeWarrior
 - It is fully compatible with Toolbox, MPW tools, and most third-party products
- (SAPROF) Our Price **\$899**



Macintosh Common Lisp 4.0 by Digitool, Inc.

Macintosh Common Lisp provides users with a rich set of object-oriented dynamic language features making it especially well-suited for rapid prototyping, custom development for business and education, scientific and engineering applications, and academic research.



BeOS Preview Release 2

by Be, Inc.

As the Internet spreads and electronic media becomes more prevalent, the high-performance needs of digital content design and the complex, aging architectures of current mainstream operating

systems are coming into conflict. The BeOS is the first operating system designed to unlock the door to much more powerful personal computers, and extract more performance from the systems we use today.

(SBEOS) Our Price **\$49**

- Power PC native environment & compiler, full Macintosh support
 - CLOS, the standard Common Lisp object system
 - Interactive dynamic environment, multiple processes
 - Automatic memory management and self-typing data
 - Ephemeral garbage collector, smaller application footprint
 - Compiles with Common Lisp industry standard and smart programmable tools, 110+ mb of user contributed code
 - Complete on-line documentation (manual sold separately)
 - Software license and registration card
- (SMCLISP) Our Price **\$675**



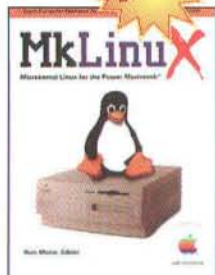
Visual Cafe for Java Professional Development Edition by Symantec

- Powerful Java and JavaBeans IDE with Plug-and-Play JavaBeans functionality
- 100+ Professional, Reusable JavaBeans Components with Source Code
- Full JDK 1.1 Support via MRJ 2.0
- Latest Java Foundation Class (JFC) Library
- Comprehensive source editor with color coding and keyword highlighting
- On-the-fly Class reusability
- Turn Auto Code Generation On/Off Environment (RRAD On/Off)
- Powerful debugging Tools including debugging in any application running MRJ 2.0 and breakpoint view
- Netscape communicator 4.0
- Full featured Visual Page HTML Authoring Tool (SVCJAVA) Our Price **\$299**



Visual Cafe for Java Database Development Edition by Symantec

- FileMaker Pro and BlueWorld Lasso trialware versions
- 100+ Database aware JavaBeans Sybase SQL
- Anywhere server-side database with development and runtime engines for Win95/NT
- DbANYWHERE server-side server 100% JDBC middleware with native drivers for Informix, Sybase, Oracle and Microsoft SQL server
- Support for over 30+ databases through ODBC
- Swift and scalable data access with dbANYWHERE's true three tier architecture
- Netscape Fast Track Web Server (Win 95)
- Netscape Communicator 4.0 (SVCJDB) Our Price **\$499**



MkLinux: Microkernel Linux for the Power Macintosh by Prime Time Freeware

MkLinux is a native port of Mach 3 and the Linux 2.0 kernel, complemented by hundreds of commands from BSD, GNU, and X11. It runs on most (NuBus and PCI bus) Power Macintosh systems; Performa,

PowerBook, and multiprocessor ports are currently under development.

MkLinux is robust, powerful, freely distributable, and source code compatible with most other Linux systems. It provides a full suite of development tools, support for AppleTalk, HFS, and Objective-C, and access to a vast amount of free software. MkLinux is a great way to "come up to speed" on Mach, UNIX, and Rhapsody.

- MkLinux user community supports FTP and web servers, development and porting efforts, and several mailing lists
- The Apple sponsored reference release contains a wealth of introductory and reference material on Linux, Mach, NeXT, and the Power Macintosh
- Includes free 3.0 upgrade (BMKLINUX) Our Price **\$49**

Order Toll-free
800-MACDEV-1
(800-622-3381)



VIP-BASIC: Visual Interactive Programming in BASIC by Mainstay

Now you can create full-featured, stand-alone Macintosh and Power Macintosh applications in standard BASIC code! VIP-BASIC 2.0 is the fastest way to program your Macintosh.

- Rapid application development environment with application framework, mix and match: VIP-BASIC high-level subprograms
- Import pre-existing BASIC code: automatically integrate BASIC code, export C Code for compiling: automatically convert your BASIC code to C for compilation with Metrowerks' CodeWarrior (SVIPBASIC) Our Price **\$195**



VIP-C: Visual Interactive Programming in C by Mainstay

Now you can create full-featured, stand-alone Macintosh and Power Macintosh applications in just minutes. VIP-C 2.0 is the first rapid application development system for creating complete Macintosh programs in standard ANSI C.

- Includes powerful, tightly integrated visual debugger, Import pre-existing C code: automatically integrate C code with a current project
- Includes full-featured mini database: (ltd to 32K) of the powerful VIP-BASIC database manager gives you everything you need to setup royalty-free, multi-user database applications (SVIPC) Our Price **\$295**



CodeBuilder

by Tenon Intersystems

CodeBuilder is a powerful and unique Macintosh software development tool for porting existing apps or developing new, advanced applications on Power Macs and Power Mac clones.

- A powerful Macintosh software development tool suite of C, C++, Objective-C, Java, Ada, and Fortran development tools
- Complete UNIX & X development environment for developing UNIX or Macintosh apps
- Includes compilers and source-code debugger for Objective C, and C, C++, Ada 95 and Fortran 77
- Web & internet scripting tools: Perl, MacPerl, tcl/tk, bash, sh, and csh
- Supports Rhapsody kernel APIs and Rhapsody TCP sockets (SMIOCODEB) Our Price **\$149**

SmalltalkAgents for Macintosh

by Quasar Knowledge Systems, Inc.

- An Integrated Development Environment (IDE) based on QKS Smalltalk.
- "Live" direct manipulation of your objects
- Dynamic, interactive and iterative development process
- Easy and full access to the features of the Mac OS(tm) and Mac Toolbox
- Link your non-Smalltalk code fragments with Code Fragment Manager (CFM) support
- Cross-reference, access, view, and manipulate your code and objects with a sophisticated database for source code management
- Includes an Application Delivery Toolkit(tm) (ADT) that allows you to create royalty-free, standalone, double-clickable applications (SSTA) Our Price **\$395**

**WAIT...
There's
More!**

Here's a list of all available products. For full product descriptions please see our Web site, or feel free to call, fax, or E-mail us.

PRODUCT	CODE	OUR PRICE
LPA MacProlog Developers Edition	SLPAD	995.00
LPA MacProlog Programmers Edition	SLPAP	495.00
LS Fortran Pro	SLSFORT	595.00
LS Fortran Plug-In	SLSFPI	199.00
Mac FORTRAN II	SFORT2	595.00
Power MachTen-UNIX	SM10PPC	695.00
Presenting Magic Cap	BPRESMAGIC	15.25
Think Pascal 4.0	SPASCAL	165.00

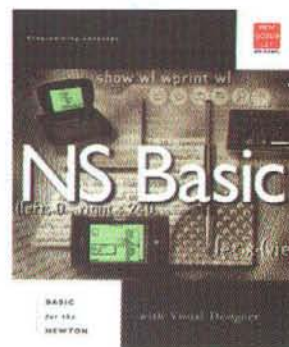
Check out our Web site!

• Full product descriptions • Hundreds of more products
<http://www.devdepot.com>

NS BASIC 3.6 for the Newton with Visual Designer

by NS BASIC Corporation

- A fully interactive implementation of BASIC programming language
- Runs entirely on the Newton — no host is required
- Create files, access the built in soups, and the serial port for input and output
- Work directly on the Newton, or through a connected Mac/PC and keyboard
- Get the **BASIC Internet Tool**, available at no charge to NS BASIC users from www.nsbasic.com
- Release Notes with sample code are available from the same location
- Runs on any Newton MessagePad 130 with NS BASIC and the Newton Internet Enabler. Also runs on MP 1201s with NOS 2.0 that have full memory available
- Write short programs to access News, mail and the web (SNSBASIC) Our Price **\$99**



ObjectMaster Professional Edition

by Altura Software, Inc.

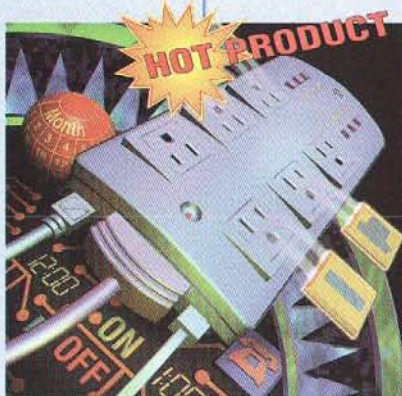
Object Master is an innovative programming environment that provides all the necessary tools to write, organize, and navigate through source code.

- Write code using the most robust source code editor available on the desktop
- Organize source code into projects to quickly access and manipulate all files
- Navigate through source code using intuitive graphical Browser windows

(SOMPE) Our Price **\$399**

PowerKey Pro Model 200 by Sophisticated Circuits

PowerKey Pro Model 200 lets you start up and shut down your Mac and up to five peripherals with a single keystroke. Two groups of switched outlets let you control some peripherals separately. PowerKey also features phone ring startup which lets you access your Mac while on the road. Powerful scheduling features let you control your outlets with "hot keys" or perform tasks unattended. Start up your computer at any time of the day or night, open applications and run AppleScripts or QuickKeys. Add the optional Server Restart Option and you can even restart crashed servers automatically! System Requirements: Mac with ADB port, System 7 or later. Telephone features require analog phone line. (HPKEY2) Our Price **\$99**



PowerKey Pro Model 600 by Sophisticated Circuits

PowerKey Pro Model 600 is "the world's smartest power strip!" Start up and shut down your Mac and peripherals with a single keystroke. Includes six individually-switched outlets, with manual switches and indicator lights. Powerful scheduling features let you control outlets with "hot keys" or perform tasks unattended. Start up your computer at any time of the day or night, open applications and run AppleScripts or QuickKeys. Complete telephone controllability lets you start up the computer, switch outlets or run complex events using custom touch-tone commands. For a limited time, Model 600 includes the Server Restart Option. Restart crashed servers automatically! System Requirements: Mac with ADB port, System 7 or later. Telephone features require analog phone line. (HPKEY6) Our Price **\$199**



NEW PRODUCT!

SpotCheck by GenieWorks, LLC

SpotCheck is a language-based editor that "knows" the Java language. It is designed to help a Java programmer produce correct code without relying on confusing and untimely feedback from a compiler. Specifically, SpotCheck identifies syntax errors and semantic errors (undefined names, type mismatches, etc.) those errors normally returned by a compiler. This analysis is performed after each edit, giving the programmer immediate feedback on errors. SpotCheck provides a host of additional features, including:

- smart links to name declarations
- cross-referenced Java APIs
- editing with popup menus
- interfaces to helper apps to compile & run
- hierarchical project browsing
- color-coded syntax

(SCHK) Our Price **\$59**

NEW PRODUCT!



Joy Explorer by AAA+ Software

Joy Explorer gets you started with Rhapsody programming quickly:

- Step by step tutorial for implementing simple applications
- Graphical application inspector for inspecting classes, instances, methods, and data structures inside any Rhapsody application.
- Interactive sending of messages to objects
- Extension to Rhapsody's Interface Builder which permits Implementation of prototypes right within Interface Builder - no need to use any tools except Interface Builder; no need to compile and link.
- Example programs with full source code, including source code for application inspector.

(SJOYEX) Our Price **\$69**

NEW PRODUCT!

Joy Developer by AAA+ Software

Joy Developer is an essential tool for anyone serious about developing for Rhapsody:

- Rapid application development right within Rhapsody's Interface Builder
- Incremental extension of existing applications (test-fix-go) without any need to recompile, relink or restart the application
- Turn interface files into standalone applications that run on any Yellow Box platform simply by using the new "Save Nib As App" command in Interface Builder
- Mix-and-match interpreted scripts and compiled code freely; anything you can do in compiled Objective-C and Java can be done in Joy - and more.
- Full support for all Objective-C classes, typedefs, macros, data structures, exceptions, Tcl 8.0 commands and extensions, including byte-code compiler to boost performance

(SJOYDEV) Our Price **\$399**



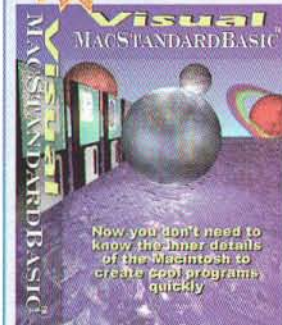
NetMinder Ethernet by Neon Software

NetMinder Ethernet is a software-only protocol analyzer which captures and decodes a full range of Ethernet protocols including IP, AppleTalk, NetWare, NetBIOS and DECnet. Features include:

- Sophisticated long-term monitoring with HTML output
- Intuitive and powerful filtering capabilities
- Automatic mapping of names to Ethernet, AppleTalk, and IP Addresses
- Rules-based engine for detecting unusual network conditions
- Customizable graphs for bandwidth utilization and packet rates

(SNETMD) Our Price **\$715**

HOT PRODUCT



Visual MacStandardBasic 3.0

by ZCurve Software

Visual MacStandardBasic is the new standard for creating both 68K and Power Macintosh applications.

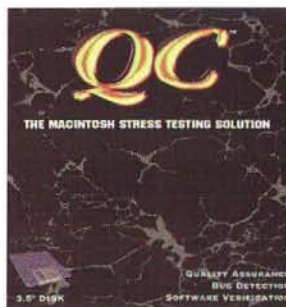
- Applications can be visually created in minutes
- Visual controls such as command buttons, text boxes, list boxes, radio buttons, check boxes, scrollbars, icons, pictures and timers can be created and modified instantly
- Use color graphics, animations, movies, sounds and speech in your programs
- Console text window option helps converting older BASIC source code from other platforms
- Online tutorial, manuals, sample projects get you programming quickly (SVMACSB) Our Price **\$29.95**

QC

by Onyx Technology, Inc.

High performance runtime stress testing for applications.

- Tests include heap checks, purges, scrambles, handle/pointer validation, dispose/release checks, write to zero, de-reference zero as well as other tests like free memory invalidation and block bounds checking
- Extremely user friendly — ideal for non-programmer testers
- Also available in Japanese (SQC) Our Price **\$99**



Web Ware

by BeachWare, Inc.

The ultimate collection of clip media and templates for building your own Web Page. An incredible selection of Shockwave movies, animated GIFs, buttons, bullets, dividers, and sample HTML pages. There are literally thousands of graphical elements on this disc, all there to spice up your web page. In all, it's about 300 megabytes of creativity only a mouse-click away! System Requirements: PC - 486 or better with 8 MB RAM, Sound card, SuperVGA, CD-ROM drive. Macintosh - Color Mac with 8 MB RAM, CD-ROM drive.

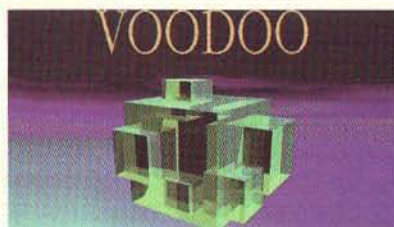
(SWEBW) Our Price **\$24**

StoneTable 68K/PPC

by StoneTable Publishing

StoneTable is a powerful and professional replacement for the List Manager used by developers worldwide. Version 3.0 is a new release with many improvements including better clipboard and drag/drop integration with other applications.

- Available for use with CodeWarrior C & Pascal
- Includes libraries for 68K (A4 & A5) and PowerPC
- An LTable-like class is provided to incorporate StoneTable into the PowerPlant environment (SSTONEFAT) Our Price **\$199**



VOODOO 1.8

by UNI SOFTWARE PLUS

- Stand-alone version control tool for all sorts of projects (software development, documentation, design, CAD, publishing, etc.)
- Smooth integration with Metrowerks CodeWarrior and BBEdit.
- Simple and clear management of variants and revisions of entire projects (not only of single files)
- Easy-to-use graphical project browser gives access to all versions that were ever stored.

Spotlight

by Onyx Technology, Inc.



Spotlight is a stand alone debugging aid that performs memory protection (arrays, heap accesses, outside your heap, low mem, etc), discipline checking on toolbox calls, and leaks detection.

- Spotlight is sold on an annual subscription basis
- The subscription service provides all updates
- Includes maintenance releases for one year after the initial purchase or renewal date. (SSPTLT) Our Price **\$199**

Celestin

Apprentice 7

by Celestin Company

Apprentice 7 is a high-quality CD-ROM collection of over 600 megabytes of up-to-date source code, utilities, and info for Mac programmers. All of the source code and utilities are completely new or updated for this release.

- Frontier 4.1, the highly-acclaimed scripting environment
- More PowerPlant AND many more PowerPC samples
- Cool new languages and environments added (Clean, Eiffel, F, Tcl-Tk)
- Hot new demos from leading Mac development companies (SAPPRENT) Our Price **\$35**

- Recording of the complete history (who made which changes when and why)
 - View differences between versions (not only for text files!)
 - Efficient delta storage of arbitrary files (text as well as non-text files) gains savings of 95 % and more
 - Administration of users with hierarchical access rights
 - Configurable local file locking (Finder flag or 'ckid' resource)
 - Scriptable, essential parts PowerPC native
- | | |
|---------------------------|---------------|
| Single license (SVOODO01) | \$229 |
| 2 pack (SVOODO02) | \$359 |
| 5 pack (SVOODO05) | \$799 |
| 10 pack (SVOODO010) | \$1369 |
| 20 pack (SVOODO020) | \$2399 |

Additional pricing available on request.

SEE RELATED CATEGORY: Dev. Environments



ObjectSet Mail SDK by Smartcode Software

- Powerful C++ classes for integrating Internet e-mail in your applications
- Helps you write software that can share mail with other leading e-mail products
- Royalty-free MIME, SMTP, and

POP3 APIs for Macintosh, Windows, and Unix

- Gives you the most robust MIME parser and encoder available
- Ideal for use in Internet and Intranet environments
- Comes complete with samples with documented, reusable source code
- Free standard technical support

(SOSMSDK) Our Price **\$495**

CodeBuilder by Tenon Intersystems



CodeBuilder is a powerful and unique Macintosh software development tool for porting existing apps or developing new, advanced applications on Power Macs and Power Mac clones.

- A powerful Macintosh software development tool suite of C, C++, Objective-C, Java, Ada, and Fortran development tools.
- Complete UNIX & X development environment for developing UNIX or Macintosh apps
- Includes compilers and source-code debugger for Objective C, and C, C++, Ada 95 and Fortran 77
- Web & internet scripting tools: Perl, MacPerl, tcl/tk, bash, sh, and csh
- Supports Rhapsody kernel APIs and Rhapsody TCP sockets (SMIOCODEB) Our Price **\$149**

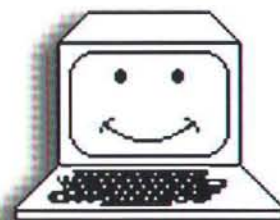
STEPUP SOFTWARE

Guide Composer™ 1.2 by StepUp Software

- Create powerful Apple Guide help systems for any new or existing Macintosh application
- Provides a WYSIWYG development environment: Guide content is developed in Guide windows
- Design topics, phrases, and panels in the same format as the user will use them
- Features are WYSIWYG interface, Topics, phrases, and hierarchical phrases, Coach marks, Fully-Integrated with Apple's Guide Maker (distributed with Guide Composer), compiles scripts automatically, PICTs in Panels, Generated Guide scripts are modifiable
- FREE Update to all registered Guide Composer users. Demo is available at <http://www.guideworks.com/> (SGCOMP) Our Price **\$99**

SEE RELATED PRODUCTS: AppleGuide Complete, Danny Goodman's AppleGuide Starter Kit, Real World AppleGuide

B-Tree HELPER 2.2 by Magreeable Software



- Inexpensive database engine for Macintosh programmers in C source code
- Uses contiguous fixed length blocks
- Expands the file as necessary and contracts files when possible
- Inserts and deletes keys in one or more B-Trees
- Finds keys equal to, less than, or greater than a given value in a few hundredths of a second
- Finds lists of records whose keys are equal to, less than, or greater than a given value or are in a range of values (SBTREE) Our Price **\$149**



BBEdit 4.5 by Bare Bones Software

BBEdit 4.5 is a powerful, easy-to-learn text and HTML editor that offers developers and HTML authors the ability to build on its core functionality to suite their

specific needs through its plug-in architecture and scripting capabilities. This new version includes: a visual table tool that speeds page and site development, contextual menu support for Mac OS 8, improved storage for 'grep' patterns, scriptable HTML authoring preferences and more. It still provides: unparalleled searching muscle with support for both 'grep' style and advanced literal searches, the ability to quickly compare differences between files or entire folders, integrated support for Symantec's IDE, Metrowerks CodeWarrior, THINK Reference 2.x, MPW Toolserver and most other environments and a heck of a lot more. (SBBEDIT) Our Price **\$119**

Also see Internet Related. page 12

Step-Up Installer Pack by StepUp Software

- Package of several Installer "atoms" that let developers incorporate graphics, sounds, file compression and custom folder icons into installation scripts
- Compression formats supported are Compact Pro & Diamond
- Each atom also available separately
- Compression requires additional licensing (SINSTALL) Our Price **\$219**

ScriptGen Pro by StepUp Software

- Installer script generator which requires no programming or knowledge of Rez
- Supports StepUp's InstallerPack, Stuffit decompression, Compact Pro decompression, custom packages, splash screens, network installs, and resource installation (SSCRIPTGEN) Our Price **\$169**

BeSpecific 3

(Third in a series)

by Adamation

The best-selling BeSpecific CDROM series provides the best in BeOS shareware across a wide range of application areas, including:

- Productivity programs
- Latest source code
- Programming tools
- Graphics Games
- Commercial demos

Newsgroup archives (comp.sys.be) Developer mailing list (Be DevTalk) BeSpecific 3 is brimming with useful BeOS Preview Release-compatible software. A "must have" companion for all users of the BeOS.

(SBESPEC) Our Price **\$39**



Pilot Attaché Disk 1

(First in a regular series)

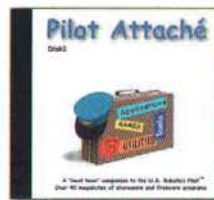
by Adamation

The Pilot Attaché CDROM, designed for the popular US Robotics Palm Pilot organizer provides you the best in Pilot shareware. Extend your Pilot's capabilities across a wide range of application areas, including:

- Personal productivity tools
- Newsgroup information
- Programming tools
- Games
- Utilities

Fully tested, Pilot Attaché's shareware treasure trove will help you get the most out of your Palm Pilot. When you travel with your Pilot, don't forget your Attaché. Pilot Attaché...your passport to success.

(SPATCHE) Our Price **\$29**



Tools Plus libraries + framework

by Water's Edge Software

Easily create compact, fast running, professional looking applications and plug-ins*. Tools Plus lets you create virtually any user interface element with a single routine, and it transparently provides a robust infrastructure to make all your pieces work together as an application.

- Simplifies programming and thins source code
- Automates all standard GUI elements
- Thousands of extras, from floating palettes and tool bars to powerful picture buttons
- Includes numerous 3D grayscale options

- Over 1/2 MB of custom fonts, icons, cursors, and other resources
 - Includes SuperCDEFs world-class controls (an \$89 value) free
- (STOOLCW) Our Price **\$249**
CodeWarrior Gold
(C/C++ & Pascal, 68K & PPC)

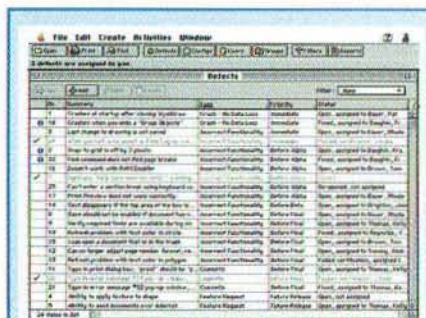
(STOOLCWB) Our Price **\$199**
CodeWarrior Bronze (C/C++ & Pascal, 68K)

(STOOLSYMT) Our Price **\$199**
Symantec (THINK) C/C++ and THINK Pascal (68K)

(STOOLSYM) Our Price **\$149**
Symantec (THINK) C/C++ (68K)

(STOOLPAS) Our Price **\$149**
THINK Pascal (68K)

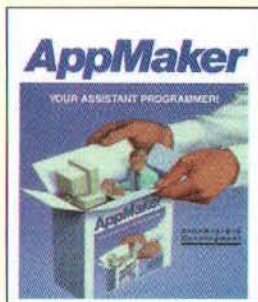
*CodeWarrior required to write plug-ins



TestTrack-Bug Tracking the Macintosh Way

by Seapine Software, Inc.

- Tracks bugs, feature requests, test configurations, users, and more
 - Includes notifications, security, a powerful filter mechanism, and multiple reports
 - Links your testers, engineers, documentations staff, and project managers together to ensure all bugs are identified, fixed, and documented
 - Eliminates the need to build custom bug tracking solutions using general purpose database tools
 - Supports single- and multi-user bug databases (additional licenses required to use multi-user features)
- (STETR) Our Price **\$169**



AppMaker

by Bowers Development

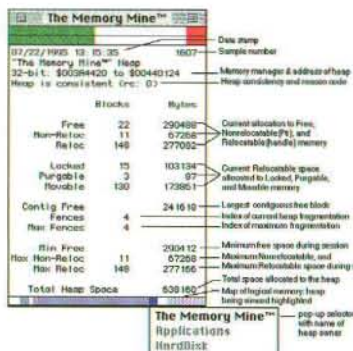
- Develop the user interface for a Macintosh application using the original interface builder
- Just point and click to design your application
- Creates resources and generates excellent source code
- Supports most development environments including Metrowerks, Symantec, or MPW; C, C++, or Pascal; procedural or object-oriented, using PowerPlant, TCL, or MacApp

- The generated code uses the Universal Headers to provide PowerMac compatibility
- Great tool for beginners to learn object-oriented and Macintosh Toolbox programming techniques
- Includes one-year subscription on CD and hardcopy documentation

(SAPPMKE) Our Price **\$199**

Check out our Web site!

• Full product descriptions • Hundreds of more products
<http://www.devdepot.com>



Memory Mine

by Adianta, Inc.

- Monitor heaps, identify problems such as memory leaks, and stress test applications
 - Active status of memory in a heap is sampled on the fly: allocation in non-relocatable (Ptr), relocatable (Handle) and free space is shown, as are heap corruption, fragmentation, and more
 - Allocate, Purge, Compact, and Zap memory lets users stress test all or part of a program
- (SMEEMINE) Our Price **\$99**

SuperAnalyst

by SuperSoft

SuperAnalyst is an easy-to-use data analysis and plotting application written specifically for the Apple Macintosh computer and PowerPC. It provides a wide range of X-Y plotting and analysis capabilities at a click of the mouse. It is easy to use and provides interactive control of the appearance of almost every characteristic of your plot. You can overlay multiple plots on the same graph, and the number of points is limited only by the memory of your computer.

(SSANAL) Our Price **\$99**

SuperPlotPRO

by SuperSoft

- Plotting data from your program was never easier
- A Plotting and Chart Library callable from Fortran, Pascal, and C
- Plot Types: Scatter, log-log, x semi-log, y semi-log, Ddouble Y, cross, line, line w/ symbols, bar, stacked bar, column, stacked column, area, pie, polar

(SSPLOTPRO) Our Price **\$295**



Future BASIC II

by Staz Software

FutureBASIC II is the award winning leader in Macintosh BASIC programming.

- Source level debugger and Interactive compiler/editor
 - Multi-file Project manager and Multi-file find and replace
 - Super fast compilation, 32 bit clean, and System 7.x savvy
 - QuickBASIC converter
 - Getting Started manual with over 500 example files
 - Full support of standard BASIC
- (SFBASIC2) Our Price **\$229**

AG Author

by Lakewood Software

AG Author 1.0 is a full-featured Apple Guide authoring tool with fully customizable project template. The following features are unique to AG Author:

- Support for styled, colored, & hot text
 - Fully customizable project template
 - Flexible compile options
 - Find & replace tool for scripts
 - Multiple open projects
 - Rapid deployment of project globals
- (SAGA) Our Price **\$99**

SEE RELATED PRODUCTS: AppleGuide Complete, Danny Goodman's AppleGuide Starter Kit, Real World AppleGuide



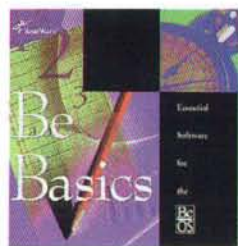
SoftPolish CD-ROM

by Bare Bones Software

- The essential tool for software quality assurance on the Macintosh

- Helps you identify inconsistencies with Apple's user interface guidelines, misspelled words, missing resources, and other mistakes
 - Provides tools to put the finishing touches on software distribution packages prior to release
 - Works independently of any programming language or environment
 - Ideal for sanity checking software throughout the development process
- (SSOFTPOL) Our price **\$99**





Be Basics

by BeatWare

Be Basics is the first productivity suite to combine the essential tools you use everyday with the incredible power of the BeOS. Be Basics offers the BeOS developer the ability to:

- Combine text, tables, graphs and pictures to create compelling documents
- Work with multiple files simultaneously without degrading performance
- View all layout, style and content changes as they happen
- Import Microsoft Excel files
- Plug-in third party graphs and filters

Be Basics requires the BeOS, 16 MB of RAM, 2 MB available hard disk space and a 256-color display adapter.

Price includes all major and minor upgrades through version 2.0 via electronic distribution.

(SBEBASIC) Our Price **\$69**

**WAIT...
There's
More!**

Here are more products. For full product descriptions please see our Web site, or feel free to call, fax, or E-mail us.

PRODUCT	CODE	OUR PRICE
Bee-one	SBEEONE	\$139.00
C-tree Plus® Database Handler	SCTPDH	\$895.00
CompileIt!	SCOMPT	\$149.00
CPU Doubler	SCPU2X	\$79.00
DesignWorks 4.0	SDWORKS	\$995.00
dtF	SDTF	\$695.00
EtherPeek	SEPEEK	\$745.00
Fortran 77 SDK	SF77	\$699.00
ICONIX PowerTools-6 Pack	SICPP6	\$5,945.00
ICONIX PowerTools-8 Pack	SICPP8	\$6,945.00
ICONIX PowerTools-10 Pack	SICPP10	\$7,845.00
ICONIX PowerTools-AdaFlow	SICADA	\$1,395.00
ICONIX PowerTools-ASCII Bridge	SICASCII	\$1,395.00
ICONIX PowerTools-CoCoPro	SICCOCO	\$1,395.00
ICONIX PowerTools-DataModeler	SICDATAMOD	\$1,395.00
ICONIX PowerTools-FastTask	SICFASTTASK	\$1,395.00
ICONIX PowerTools-FreeFlow	SICFREEFL	\$1,395.00
ICONIX PowerTools-Object Modeler	SICOBJMOD	\$1,395.00
ICONIX PowerTools-PowerPDL	SICPOWER	\$1,395.00
ICONIX PowerTools-QuickChart	SICQUICKCH	\$1,395.00
ICONIX PowerTools-SmartChart	SICSMART	\$1,395.00
ICONIX Training & Consulting	TICONIX	\$2,945.00
IMSL Math and Stat Library	SIMSLSTAT	\$495.00
Info-Mac X	SINFOMAC10	\$39.00
Ionizer Real-Time Spectral Reshaping Tool	SIONIZER	\$800.00
LiveAccess™ 1 User Edition	SLAUE	\$69.00
LiveAccess™ 1 Developer Edition	SLADE	\$99.00
LiveCard	SLCARD	\$149.00
LJ Profiler	SLJPROF	\$295.00
MacA&D 6.0	SMACADP	\$1,995.00
MacFlow™: Flowchart Design and Development	SMACFLO	\$179.00
Mac Source II	SMACSOURCE	\$29.95
Nisus Writer 5.0	SNISUSW	\$220.00
Plan & Track™: Project Planning and Management	SPLNTRK	\$179.00
Phyla™: Object-Oriented Database	SPHYLA	\$179.00
QUED/M 3.0	SQUEDM	\$89.00
r-tree Report Generator	SRTRG	\$445.00
Spellswell Plus 2.1	SSPELL	\$49.00
SuperPlot	SSPLOT	\$195.00
Visual Café	SVCAFEMAC	\$199.00
VText	SVTEXT	\$349.00

OpenGL for the Macintosh

by Conix Graphics



OpenGL is the premier 3D graphics library that allows software developers the ability to develop high-quality, interactive 2D and 3D graphics applications. OpenGL can perform the following wide range of functions which will enhance the development of all graphics software:

- Geometric primitives (points, lines, and polygons)
 - RGBA or color index mode
 - Viewing and modeling transformations
 - Texture Mapping, Lighting, Shading and Z Buffering
 - Atmospheric Effects (fog, smoke, and haze)
 - Alpha Blending (transparency)
 - Antialiasing, Accumulation Buffer, Stencil Planes
 - Display list or immediate mode
 - Polynomial Evaluators (to support Non-uniform rational B-splines)
 - Feedback, Selection, and Picking Raster primitives (bitmaps and pixel rectangles)
 - Pixel Operations (storing, transforming, mapping, zooming)
- (SOPENGL) Our Price **\$389**

HOT PRODUCT

NEW VERSION



Lasso 2.5

By Blueworld Communications

APRIL 1997
Macworld **★★★★** 72

Lasso 2.5 is the ultimate FileMaker Pro Web development tool for creating online stores, discussion groups and other robust web applications that can handle hundreds of thousands of

hits per day. Over 50 new tags including math, string and variable allow unparalleled data handling capability. Create embedded operations with the inline command. Lasso Server edition supports multihoming and offers superior performance.

Order Lasso today the award-winning tool that which sets the standard for Mac OS dynamic Web database publishing.

- Multi-threaded performance
- Supports CDML and LDML
- Enhanced security for electronic commerce

Lasso 2.5 CGI/Plug-In (SLASPG) **\$499**

Lasso 2.5 Server (SLASSVR) **\$649**

HOT PRODUCT



BBEEdit 4.5

by Bare Bones Software

(SBBEDIT) Our Price **\$119**

Also see Tools, Libraries and Utilities, page 8

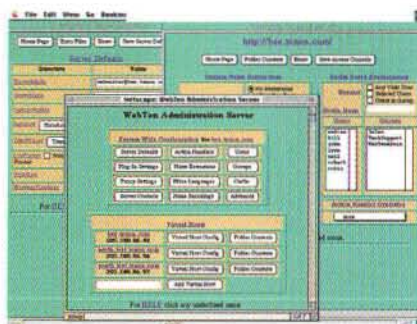
WebTen

by Tenon Intersystems

WebTen is an industrial-strength, high-performance Apache Web server for Power Macs. WebTen's Web-based browser interface enables local or remote administration via your favorite browser. Since Apple's NeXT acquisition, Tenon has extended their unique "UNIX virtual machine" technology to produce a set of "Rhapsody-Ready" internet applications. WebTen is the first offering in this series.

- WebTen is the fastest Web server on Power Macintosh
- Sustains up to 10,000 connections a minute, or over 10 million connections a day
- Apache runs in Tenon's multi-threaded, pre-emptive multitasking environment
- Tenon's unique technology supports the widely acclaimed Apache Web server as a double-clickable Macintosh application

(SWEETEN) Our Price **\$495**



NEW PRODUCT!



CyberStudio

by GoLive Systems

GoLive CyberStudio is the complete solution for HTML layout, design and Web site management. It gives Macintosh users unprecedented creative control and flexibility when designing a Web site. GoLive

CyberStudio lets graphic designers and publishers visually design and manage a professional-quality Web site—including the latest multimedia features—without performing any HTML programming. Yet, GoLive CyberStudio also includes HTML source-code and JavaScript tools, which help Web designers and programmers integrate interactivity into a site. Because GoLive CyberStudio is the first Web site design software to always work in a native HTML file format, it allows graphic designers and Web programmers to work well together.

(SCYBERS) Our Price **\$349**



ObjectSet Mail SDK

by Smartcode Software

- Powerful C++ classes for integrating Internet e-mail in your applications
- Helps you write software that can share mail with other leading e-mail products
- Royalty-free MIME, SMTP, and POP3 APIs for Macintosh, Windows, and Unix

- Gives you the most robust MIME parser and encoder available
 - Ideal for use in Internet and Intranet environments
 - Comes complete with samples with documented, reusable source code
 - Free standard technical support
- (SOSMSDK) Our Price **\$495**

HyperGuide 1.0

by Lakewood Software

HyperGuide 1.0 is a hybrid multimedia authoring tool and on-line documentation system for the Macintosh and World Wide Web. HyperGuide provides integrated searching, indexing and bookmarking features.

Supported media elements include: rectangle and scrolling fields, lines and shape fills, most QuickTime-supported image formats, anti-aliased text and QuickTime VR movies. HyperGuide also includes an integrated screen capture utility and user-configurable slide show mode.

(SHYPGUD) Our Price **\$149**



PageCharmer: Sizzling Effects...



PageCharmer 1.0

by Mainstay

PageCharmer is a set of customizable interactive applets that enhance web pages without writing a single line of HTML code. Whether the web site is already up and running or designing one from scratch, PageCharmer gives you the power to make it stand out from the crowd with sophisticated applets that can be personalized to fit most any need.

FEATURES:

LiveG-Map, LiveT-Map, LiveG-Button, LiveT-Button, LiveGT-Button, LiveG-Ticker, LiveT-Ticker, LiveG-Marquee, and LiveT-Marquee.

(SPGCHRM) Our Price **\$99**

Power MachTen 4.0.3

by Tenon Intersystems

MachTen is the only Macintosh product that can turn your Macintosh into a complete Unix workstation. Based on BSD4.4 and the Mach kernel, MachTen brings the power of Unix to your desktop at an extremely attractive price point. MachTen enables you to:

- Run a high speed internet server, complete with WWW, FTP, NFS, DNS and print service
- Build a Multihomed Web Server
- Develop applications in a Unix development environment, replete with the acclaimed GNU development toolset
- Program in Ada, C, C++, Pascal, Fortran, and more
- Run Xwindows applications, from remote workstations or on your Macintosh
- Run hundreds of Unix applications, already ported for MachTen and available on our Ported Applications CD-ROM
- Run Software.com Inc's acclaimed Post.Office mail transport service (SM10PPC) Our Price **\$695**



Rumpus

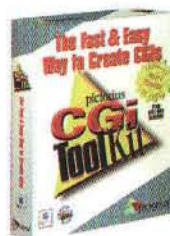
by Maxum Development

Maxum's new, high-performance FTP server for the MacOS. Based on

Maxum's RushHour TCP/IP implementation, Rumpus 1.0.1 offers the performance and reliability of high-end workstations with the ease of use, security, and flexibility of the Macintosh.

- Simplified setup, with no need to configure AppleShare, File Sharing, or Users & Groups for simple anonymous FTP
- Anonymous and/or secure server access, with separate security settings for anonymous vs. secure users
- Automatic MacBinary and Binhex encoding
- Complete logging, with separate anonymous and secure access logs, including anonymous user passwords
- Up to 32 simultaneous connections

(SRUMP) Our Price **\$195**



CGI Toolkit

by Pictorius, Inc.

The Pictorius CGI Toolkit is the fast and easy route to high performance CGIs and ACGIs for your Mac Web site.

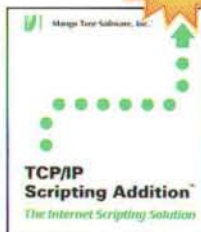
- Interactively develop CGIs while the web server, the CGI Toolkit and the browser are running on the same machine
- Interactively develop, test and debug CGIs before compiling
- Powerful debugger allows you to edit code, roll back, code and change input values while your application is running
- Fully object oriented so you can re-use your code
- Automatic handling of Apple Events so you can concentrate on building functionality
- Easy creation of multi-function CGIs which reduces application footprint and RAM usage

(SCGITLKT) Our Price **\$149**

**WAIT...
There's
More!**

Here are more products. For full product descriptions please see our Web site, or feel free to call, fax, or E-mail us.

PRODUCT	CODE	OUR PRICE
OOFIE Reporter Writer	SOORW	\$499.00
ScriptDemon	SSDEMON	\$949.00
WebAlias 1.0	SWEBALS	\$129.00
WebSiphon	SWSIPHON	\$495.00

HOT PRODUCT!

TCP/IP Scripting Addition

by Mango Tree Software

- Award-winning AppleScript scripting addition
- Allows you to write scripts using MacTCP™ commands in AppleScript™
- Send e-mail or files through a script,

check if users are logged on (via Finger), automate FTP, Gopher, NetNews, Telnet, and LPR, verify links in HTML documents, and quickly write many other TCP/IP client-server programs

- Works with AppleScript, MacTCP 2.0.4 and Open Transport (STCP) Our Price **\$49**



Scripter 2.0

by Main Event Software

For professionals, for novices, for webmasters, for solutions providers, there's only one serious choice. Scripter!

- Scripter and FaceSpan work together: one click opens your FaceSpan script in Scripter, another sends it back

- Debug handlers without modifying your scripts using the Call Box
- Applet simulation, live editing, Object map, associated terminology
- Search backwards, block generators, more navigation shortcuts, more drag-and-drop, and an even more enhanced trace log
- Now Includes ScriptBase; stores your data and media elements and share them between scripts all with a special new browser
- Easily write and compile scripts that have handler declarations and other vocabulary specific to a particular scriptable application
- Scripter is the natural companion to AppleScript for users at all levels of proficiency. Don't write scripts without it!

(SSCRIPTER) Our Price **\$199**

DynaMorph 1.5

by Morph

DynaMorph

DynaMorph is the only cross-platform, server-side scripting language. Easily build and maintain dynamic websites and web-based applications. Access external databases, separate the format of a website from its content, conduct e-commerce transactions and more. DynaMorph makes sites and applications completely portable.

(SDYNA) Our Price **\$399**

**WAIT...
There's
More!**

Here are more products. For full product descriptions please see our Web site, or feel free to call, fax, or E-mail us.

PRODUCT

PreFab Player

CODE

SPLAYER

OUR PRICE

\$95.00

NEW PRODUCT!

FaceSpan v3.0

by Digital Technology International

FaceSpan is a cutting edge interface design and rapid application development (RAD) tool which gives you the power to build and customize Macintosh applications quickly and easily.

- Acts as your front end for AppleScript or any other OSA (Open Scripting Architecture) language.
- Allows you to automate often-repeated tasks, customize and integrate existing applications, build new applications and personalize your computing environment.
- NEW! Allows you to create interfaces and applications that conform to the Mac OS 8 look and feel.
- NEW! Supported display objects now include tab panels, disclosure triangles, bevel buttons and more.
- NEW! FaceSpan run-time now launches up to 5X faster.
- Includes an unlimited, royalty-free distribution license for the interfaces and applications you create.

(SFACESPAN) Our Price **\$149**

Script Debugger

by Late Night Software Ltd.

- A powerful and flexible AppleScript authoring tool – get the most from AppleScript!
- Advanced debugging environment offers single-step script execution with breakpoints
- Script Debugger dictionary browser features a graphical view of objects provided by scriptable applications
- Includes Late Night Software Scripting Additions – a collection of more than 70 new AppleScript commands, and Scheduler, a utility that allows you to launch scripts at pre-determined times

(SDEBUG) Our Price **\$129**

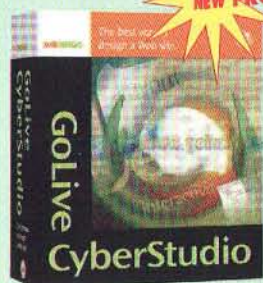


WindowScript

by Royal Software, Inc.

WindowScript is the ultimate tool for designing Macintosh user interfaces using HyperCard. Design Real "Macintosh" user-interfaces right inside HyperCard. Until now you either created HyperCard stacks or Macintosh applications. With WindowScript you can literally bring the look and feel of a real Macintosh user-interface to HyperCard. If you're a HyperCard developer, interface designer, application developer, program manager or tester searching for a prototyping tool, WindowScript is perfect for the job.

(SWSCRIPT) Our Price **\$149**



NEW PRODUCT!

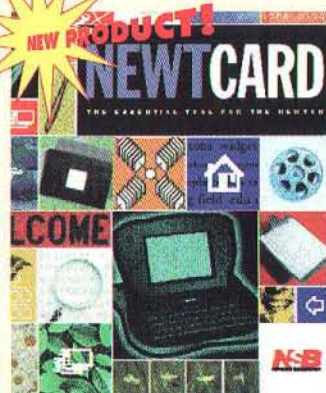
CyberStudio

by GoLive Systems

GoLive CyberStudio is the complete solution for HTML layout, design and Web site management. It gives Macintosh users unprecedented creative control and flexibility when designing a Web site. GoLive

CyberStudio lets graphic designers and publishers visually design and manage a professional-quality Web site—including the latest multimedia features—without performing any HTML programming. Yet, GoLive CyberStudio also includes HTML source-code and JavaScript tools, which help Web designers and programmers integrate interactivity into a site. Because GoLive CyberStudio is the first Web site design software to always work in a native HTML file format, it allows graphic designers and Web programmers to work well together.

(SCYBERS) Our Price **\$349**



NEW PRODUCT!

NewtCard

by NS BASIC Corporation

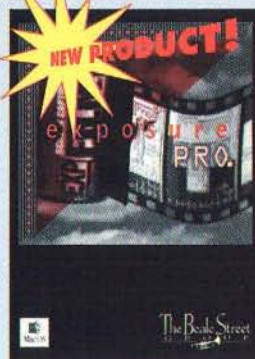
NewtCard lets you put text, drawings, pictures and sound into a stack of smart cards on your Newton. Add buttons to navigate, fields to collect data, and scripts to bring your project alive with the tap of a pen!

- Create business solutions, education courseware, interactive presentations and more

in an easy to use environment.

- Enter text, numbers dates and times easily.
- Create your own drawings or copy drawings and text from NewtWorks
- Navigate to other cards, stacks or applications by clicking buttons you create.
- Add scripts to your stacks in easy to use BASIC!

(SNEWT) Our Price **\$99**



NEW PRODUCT!

ExposurePro

by The Beale Street Group, Inc.

ExposurePro allows you to, at the touch of a key (user-defined, of course), freeze the screen, select a portion with the powerful selection tools, and save the image to a TIFF, GIF, PNG, JPEG, PICT, or PICT clipping file. You can also save to the Clipboard, Scrapbook, or even send it directly to your printer.

The power of ExposurePro lies not only in its screen capture capabilities, but also in the vast array of editing tools: painting tools, drawing tools, all with customizable features rivaling a full-blown image editing program. Convert to a different bit depth. Dither. The text tool might make you trash Illustrator!

(SEPRO) Our Price **\$119**



NEW PRODUCT!

Widgetizer

By Roundabout Logic

Widgetizer is Roundabout's peerless rapid object editor for QuickTime VR. Widgetizer generates interactive QTVR

object movies which look in at a center point or plane, simulating the effect of holding an object in the hand and turning it around to see all sides, top and bottom.

- Definable hot spots
 - Custom backgrounds
 - Frame-based animation
 - Sound integration
 - Small memory footprint
 - Clean user-friendly interface
 - Plug-in architecture
 - Exports both QuickTime VR 1.0 and 2.0
 - Photoshop Acquire Module that supports all devices including film cameras, (through scanners or photo CDs), digital cameras, video cameras and digital video cameras.
 - Supports device (mechanism) software controlled QTVR object rigs
 - Expensive software controlled mechanisms are not required.
- System Requirements: Runs on 68K or Power PC; requires allocation of 8 mb real RAM.

(SWIDG) Our Price **\$169**



NEW PRODUCT!

Nodester

by Roundabout Logic

Nodester is Roundabout's renowned rapid panoramic editor for QuickTime VR.

Nodester generates interactive QTVR panoramic movies which look out from a center or nodal point, allowing the viewer to experience the effect of a full 360 degree panorama.

- Hot spot editor
- Built-in image editor

- Photoshop Acquire Module that supports all devices including film cameras, (through scanners or photo CDs), digital cameras, video cameras and digital video cameras.
 - Small memory footprint
 - Clean user-friendly interface
 - Plug-in architecture
 - Export both QuickTime VR 1.0 and 2.0
- System Requirements: Runs on Power PC only; requires allocation of 8 mb real RAM.

(SNODE) Our Price **\$169**



QuickTime VR Authoring Studio is Now Available for only \$299 when you purchase selected Kaidan products.

Get the QTVRAS for only \$299 when you purchase the KiWi+, QuickPan Magnum or Magellan QC. This is a savings of approximately \$95 off the standard retail price. Sorry, upgrades from the original VR Suite are not available.

(HQTKB) Our Price **\$299**

Available only with Kaidan products KiWi+, QuickPan Magnum and Magellan QC.



Magellan QC by Kaidan

The Magellan QC is capable of handling objects as large as six inches in diameter and five pounds in weight, the Magellan QC is the perfect choice for those needing to capture small objects at a reasonable price. Real-world objects can be turned into 3-D virtual reality movies using the QuickTime VR Authoring Studio and the Magellan QC.

The Magellan QC leverages the capabilities of the Connectix™ Color QuickCam™ digital camera for QTVR object capture. The Color QuickCam's close focusing capability (one inch to infinity), 640 x 480 resolution, serial interface (no video card required), 24-bit color support, convenient size and low cost make it an ideal camera for many simple QTVR object movies. Using the Magellan QC is easy. Simply locate the object on top of the adjustable pedestal, perhaps with a small piece of double-sticky tape, and then adjust the arms and pedestal so that the center of the object is centered in line with the camera and the rotation axis of the swingarm.

(HMAGQC) Our Price **\$299**

Magellan Accessories

Magellan QC Pedestal Set

Two extra pedestal tube assemblies, one 2.5" and another 6" long. These extra pedestal tubes are used to support objects of varying sizes on the Magellan QC.

(HMAGPED) Our Price **\$39**

Magellan QC Detent Wheels

A pair of optional detent wheels (Color = Gold) with 8 (45 deg), 12 (30 deg), 14 (25.7 deg), 16 (22.5 deg) and 18 (20 deg) settings. The standard wheels (Color = Aqua) provide 10, 15, 20, 24 and 36 positions.

(HMDWHLS) Our Price **\$74**

QuickPan Magnum by Kaidan

The QuickPan Magnum Series consists of two models, the QPX-1 and QPX-2. Featured on both models is the new QPU-2 camera bracket. Based on the highly successful KiWi, it provides a sturdy, collapsible system for the mounting and adjusting of a wide variety of cameras and camcorders. The new base designs used on the Magnums are a refinement of our earlier bases, with the QPX-1 having a fixed base and the QPX-2 having a new low-profile micro-tilt adjustment stage. The easily adjustable click-stops will let you capture a panorama in a few seconds. The QPU-2 has two accessories, a Landscape Bracket for positioning the camera in the landscape orientation (QPLB-1) and a Counterweighting Kit (QPCW-1) used to balance large cameras or camcorders, such as the Sony VX-1000, that have a center of mass well behind the pivot axis.

QuickPan Magnum-1 (HQPMAG1) Our Price **\$499**

QuickPan Magnum-2 (HQPMAG2) Our Price **\$549**



QuickPan Magnum Accessories

Counterweighting Kit

The Counterweighting Kit includes a weight and adjustable arm that is used to offset the weight of large, heavy cameras and camcorders.

(HWHTKT) Our Price **\$129**

Detent Wheel

Detent Wheel (5-inch) (Color = Purple): 10, 14, 18, 24 and 30 Position (QPDD-2)

(HQDWHLS) Our Price **\$49**

QuickTilt Leveler

A leveling stage, similar to the one found on our QuickPan Magnum QPX-2, that mounts between your panhead or camera and your tripod. It makes the leveling process quick and easy. Particularly useful when you plan to shoot a number of QTVR/VR nodes in a short period of time.

(HQTILVR) Our Price **\$149**

HOT PRODUCT

KiWi

by Kaidan

The KiWi™ is the most affordable VR/QTVR panhead, bringing digital photographic panoramas to an even wider audience. It's the perfect companion to programs such as QuickTime VR Authoring Studio, PhotoVista and Nodester, providing a complete solution for anyone interested in adding VR panos to their websites and multimedia applications. The KiWi™ consists of two intersecting black anodized aluminum struts that adjust and lock to accommodate a wide range of cameras, such as the Apple QuickTake 100/150/200, Kodak DC50/120, APS film cameras and 35mm SLRs equipped with wide-angle lenses. KiWi™ attaches to any standard tripod and camera equipped with a standard 1/4-20 mounting thread.

(HKIW1) Our Price **\$99**



HOT PRODUCT

KiWi+

by Kaidan

The KiWi+ adds a compact, yet durable click-stop mechanism and the same twin-axis bubble level found on the top-of-the-line QuickPan Magnum Series heads. The twin-axis bubble level (recommended by Apple and VR professionals) provides a clear indication of level, even when the unit is slightly above eye level. The click-stop mechanism uses easily replaceable detent discs, which are available in a number of positions (8, 12, 16, 18, 20). The KiWi+ ships with one disc of your choice and extra discs are available separately or as a set. The click-stops speed the process of shooting a panorama by eliminating the need for the photographer to look at the unit in order to visually align the index increment.

KiWi + Ships with detent disc of your choice

KiWi+ with detent disc size 8	(KIWI8)	Our Price	\$249
KiWi+ with detent disc size 12	(KIWI12)	Our Price	\$249
KiWi+ with detent disc size 16	(KIWI16)	Our Price	\$249
KiWi+ with detent disc size 18	(KIWI18)	Our Price	\$249
KiWi+ with detent disc size 20	(KIWI20)	Our Price	\$249



KiWi and KiWi+ Accessories

QuickTilt Leveler

A leveling stage, similar to the one found on our QuickPan Magnum QPX-2, that mounts between your KiWi or KiWi+ and your tripod. It makes the leveling process quick and easy. Particularly useful when you plan to shoot a number of QTVR/VR nodes in a short period of time.

(HQTLLVR) Our Price **\$149**

KiWi-to-KiWi+ Upgrade

Includes the necessary parts required to turn your KiWi into a KiWi+ — adding click-stops and the twin-axis bubble level. Comes with a detent disc of your choice (8, 12, 16, 18 or 20 positions).

(HKIWIUP) Our Price **\$199**

KiWi+ Detent Discs

KiWi+ Detent Discs are available singly or in a set of four. In both cases you get to choose whichever discs you need.

(HDTDISC) Our Price **\$24.95** each

(HDTDISC4) Our Price **\$89** set of four

Choices include: 8, 12, 16, 18, or 20 position detent disc

Landscape Bracket

The Landscape Bracket is a right angle bracket that allows you to mount the KiWi or KiWi+ upright camera bracket in a horizontal orientation. This bracket is primarily used for cameras that have a limited field of view and you need to limit the number of shots.

(HLDBRAC) Our Price **\$42**

Flash Hotshoe Level

A dual-axis bubble level that slides into your camera's hotshoe. It's a useful tool to help level the camera on the upright camera bracket.

(HFLASH) Our Price **\$39**

Offset Spacer

The Offset Spacer is a circular spacer that may be required for very narrow cameras (i.e. certain Ricoh digital cameras) in order to position the center of the lens over the pivot axis.

(HOFFSPAC) Our Price **\$24.95**

Media Cleaner Pro

by Terran Interactive

Use Media Cleaner Pro 2.0 to optimize and compress video for CD-ROM, kiosk, or the Internet. Media Cleaner Pro automates your work flow allowing you to get the highest quality video, faster and easier than any other program on the market.

- Includes Adobe Premiere Export module
- Optimal palette generation, Drag-and-drop batch processing
- RealMedia, VDOlive and improved QuickTime support
- Dynamic Preview Window, the Media Wizard, multiprocessor support and more!

System Requirements:

68040 Mac or better (PowerPC strongly recommended, req'd for RealMedia), QuickTime 2.0 or later (2.5 strongly recommended)
8 Mb application RAM, MacOS 7.0.1 (7.5 or later recommended)
SoundManager 3.2, CD-ROM Drive
(SMCP) Our Price **\$359**

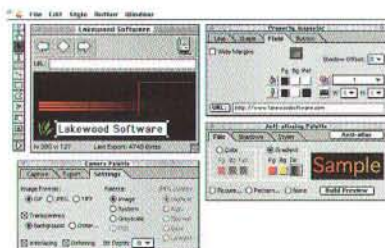
Order Toll-free
800-MACDEV-1
(800-622-3381)

webAlias 1.0

by Lakewood Software

webAlias 1.0 is an integrated image map editor and anti-aliasing text tool for web and graphic designers. Use webAlias to create complete web sites, single web pages, and graphic content for multimedia and web design projects. webAlias integrates support for line, shape, free form, field and button objects. webAlias' anti-aliasing features include support for embedded pictures and gradients in text, as well as multiple shadow and highlight effects.

(SWEBALS) Our Price **\$129**



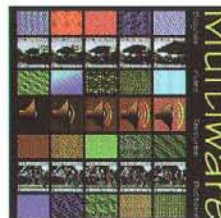
Music Tracks

by BeachWare, Inc.

A new PC/Mac & Audio multimedia music CD-ROM. The clips include musical introductions, fanfares, background music, and more. This collection offers you 100 music clips stored in .WAV format for

Windows, SoundEdit & AIFF formats for Macintosh and as Audio tracks for audio CS players. All of the music clips are completely license and royalty-free!! Mac System requirements: Mac Plus or greater, CD-ROM drive. PC system requirements: Windows 3.1 or later, Sound Blaster compatible board, CD-ROM drive.

(SMT) Our Price **\$24.95**



MultiWare

Multimedia Collection

by BeachWare, Inc.

Introducing a new Audio multimedia music CD-ROM for the Macintosh. This disc is a collection of clips ideal for Desktop Presentations and other Multimedia

applications. This incredible collection of license-free media clips is bursting with 240+ color pictures and backdrops (PICT), 200+ sound & music clips (SoundEdit), 140+ QuickTime movies, and a variety of multimedia tools for use with the Macintosh.

(SMWMC) Our Price **\$24.95**

HyperGuide 1.0

by Lakewood Software

HyperGuide 1.0 is a hybrid multimedia authoring tool and on-line documentation system for the Macintosh and World Wide Web. HyperGuide provides integrated searching, indexing and bookmarking features. Supported media elements include: rectangle and scrolling fields, lines and shape fills, most QuickTime-supported image formats, anti-aliased text and QuickTime VR movies. HyperGuide also includes an integrated screen capture utility and user-configurable slide show mode.

(SHYPGUD) Our Price **\$149**



**WAIT...
There's
More!**

Here are more products. For full product descriptions please see our Web site, or feel free to call, fax, or E-mail us.

PRODUCT	CODE	OUR PRICE
AudioTrack	SAUDIOTRK	\$270.00
Be Studio	SBESTUD	\$99.00
Captivate 4.6: Essential Graphics Utilities	SCAPTIV	\$79.00
Clip VR	SCLIPVR	\$89.00
Media Cleaner Pro	SMCPUP	\$359.00
Screen Machine	SSM	\$24.95



Power3D by Techworks

The power of an arcade on your PowerPC Based on award winning 3Dfx Voodoo Graphics. The Power3D works with your existing graphics

card and multi-sync monitor to provide you the absolute in 3D performance. Install the Power3D in your PowerPC (requires one available PCI slot in your system) and use the provided pass-through cable to turn your PowerPC into a Power Arcade system!

Power3D comes bundled with these awesome 3D enabled games:

- Quake®: Episode 1 (8 level) by Id Software
- MechWarrior® 2 by Activision
- VR Soccer™ by VR Sports (Actua™ for Europe)
- Weekend Warrior™ by Bungie

(SPWR3D) Our Price **\$249**



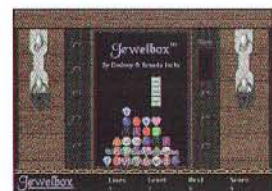
Abuse by Bungie Software

Abuse is 360 degrees of side-scrolling action. Run, jump, fall and fly in any direction - through industrial corridors, caverns and sewers. Destroy enemies in any direction with grenade launchers, rocket launchers, napalm and nova spheres! Avoid deadly traps with jet packs and turbo boost!

Key Features:

- Point and Kill Interface. Move and annihilate mutants in complete 360° freedom
- Blast your way through floors, walls and ceilings in search of the ultimate power-up!
- Abuse is 360 degrees of side-scrolling action. Run, jump, fall and fly in any direction - through industrial corridors, caverns and sewers

(SABUSE) Our Price **\$51**



1000 Games for Macintosh by BeachWare, Inc.

The best Macintosh game disc in the entire world, this CD-ROM contains over one thousand great shareware and public domain programs. Battle ugly aliens, blast apart run-away asteroids, deal yourself that royal flush or solve that 3-D puzzle, this disc has it all! System requirements: Mac Plus or greater, CD-ROM drive, and 2 MB of available RAM (4 MB of RAM when running under System 7).

(STGM) Our Price **\$24**

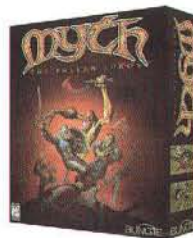
**WAIT...
There's
More!**

Here are more products. For full product descriptions please see our Web site, or feel free to call, fax, or E-mail us.

PRODUCT	CODE	OUR PRICE
A Zillion Sounds	SAZS	\$24.00
Casino!	SCAS	\$24.00
Night Sky Interactive	SNSI	\$24.00
Trivia Warehouse 2000	STW2K	\$24.00

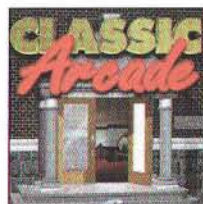
Myth The Fallen Lords by Bungie Software

PC GAMES MAGAZINE'S, Most Anticipated New Game Award



The Fallen Lords is a fully 3D real-time strategy game of epic battle. A multimetric game, Myth: The Fallen Lords gives gamers unprecedented freedom to view their forces, orbiting around the heads of a formation or zooming in for a close-up on savage melee. Myth: The Fallen Lords includes maps designed for networking, and alternate networking scenarios like Assassin and King of the Hill.

(SMYTH) Our Price **\$49**



Classic Arcade by BeachWare, Inc.

Ten of your favorite coin-arcade games, redone with killer graphics and sounds! Walk through a virtual arcade and test your game playing skills with these exciting arcade classics. Ten games, including Moon Lander, Astro-Boing, Hyper

Hockey, Ballistic Avenger, and more. System Requirements: Mac — Color Mac with 8 MB RAM, CD-ROM drive. PC 486 with 8 MB RAM, Sound card, SuperVGA, CD-ROM drive.

(SCLA) Our Price **\$24**



Marathon Trilogy Box Set by Bungie Software

The Marathon Trilogy Box Set brings all three Marathon games together in one affordable package, with tons of extras thrown in. Besides Marathon, Marathon 2: Durandal and Marathon Infinity, you'll also receive a staggering 1200 maps, featuring never-released Bungie maps and the winners of the Infinity

Mapmaking Contest, The Marathon Scrapbook (a behind-the-scenes look at themaking of the Marathon games), Marathon collectables like the Marathon 3-sticker set, and to top it off, the award-winning game that laid the groundwork for Marathon: Bungie's breakthrough Pathways Into Darkness. The Marathon Trilogy Box Set is native to the Power Macintosh, utilizes the graphics acceleration of 630 and 6200 machines, is 8, 16 and 24-bit color capable, and can be played with joysticks and game pads. The package requires a 68040 or higher Macintosh, CD-ROM drive, 8-bit color monitor (13" recommended), and System 7 or later.

(SMTBS) Our Price **\$65**



GeekWare™

You live it, you breath it... you might as well wear it! (XL only)



Alien T-Shirt

(AALIEN) Our Price **\$9.95**

Arnold T-Shirt

(ACWARNLD) Our Price **\$9.95**

Arnold Jr. T-Shirt

(AARNOLDJR) Our Price **\$9.95**

Blood, Sweat & Code T-Shirt (SS)

(ACWSBLOOD) Our Price **\$9.95**

Blood, Sweat & Code T-Shirt (LS)

(ACWLBLOOD) Our Price **\$14.95**

CodeWarrior Baseball Cap - Black

(ACWBHAT) Our Price **\$14.95**

CodeWarrior Sweatshirt - Black

(ACWSWEAT) Our Price **\$29.95**

CodeWarrior Hawaii Five-0 Shirt

(ACWHAWAII) Our Price **\$9.95**

CodeWarrior Winter Hat

(AWINHAT) Our Price **\$14.95**

Debugger Boxer Shorts

(ADBOXER) Our Price **\$16.95**

Discover Programming T-Shirt

(ADISCPT) Our Price **\$9.95**



Podeum Sport

by Rach, Inc.

Now you can have laptop stability, drop protection & mobility. If you're looking for an inexpensive, non-technical gift for a laptop owner – look no further.

- Strap your laptop to your leg
- Universal size - fits all laptops and all legs (13" - 46")
- Velcro Velcoins attach your laptop to the podeum
- Podeum allows working angles up to 90 degrees
- Dropped laptops account for 41% of all laptop fatalities
- Manufacturer's lifetime warranty.

(APODSP) Our Price **\$39**



MacTech® Mouse Pad

Slide on this! With an extra-large surface (11" by 10") and a deluxe sleek plastic coating, you'll be zooming across your screen in no time at all. Speed limit not enforced!

(AMTPAD) Our Price **\$8.95**

MacTech®

M A G A Z I N E

MacTech® Magazine

MacTech keeps Mac programmers & developers up to date with everything they need to know about software development. Topics like Rhapsody, Java, QuickTime, OPENSTEP, Objective-C, C/C++, Object Oriented Technologies, product reviews and much more!

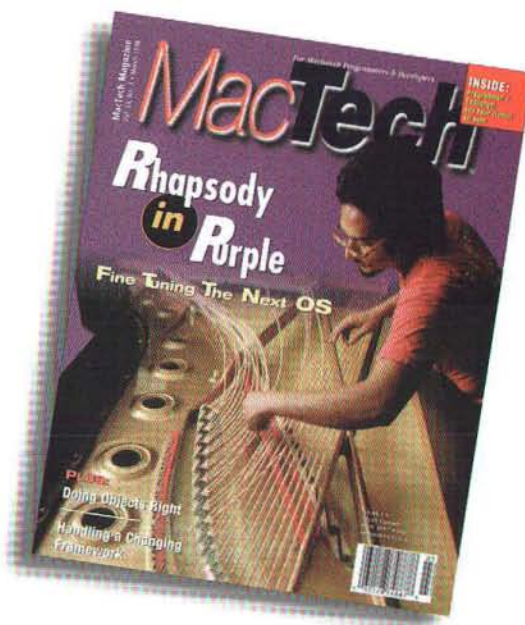
Subscriptions:

(MTYRDM) US/Domestic for 12 issues **\$47**

(MTYRCM) Canadian for 12 issues **\$59**

(MTYRFM) International for 12 issues **\$97**

Back Issues: each plus shipping (subject to availability) **\$10**

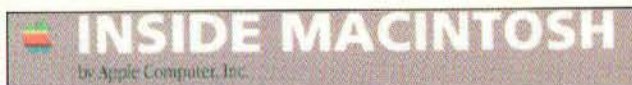


MacTech® CD-ROM Volumes 1-12

- Includes Apple's *developer* issues 1-29 (1990-1997)
- Almost 1600 articles from all 139 issues of MacTech Magazine (1984-1996) and through may of 1997
- Improved hypertext, improved indices, and a new THINK Reference Viewer – for lightning quick access!
- New hyperlinks between articles
- 100+ MB of source code — use them in your applications, with no royalties!
- Full version of THINK Reference™ — the original online guide to Inside Macintosh, Vols. I-VI
- 80MB of FrameWorks/SFA archives and the most complete set of FrameWorks archives known
- Sprocket™! MacTech's tiny framework that compiles quickly and supports System 7.5 features
- The best threads from the Macintosh programmer newsgroups plus thousands of notes, tips, snippets, and gotchas
- Popular tools that Macintosh programmers use to increase their productivity and much more!

(SMTCD12) Volumes 1-12 Our Price **\$129**

(SMTCD12U) Upgrade from any previous version Our Price **\$49**

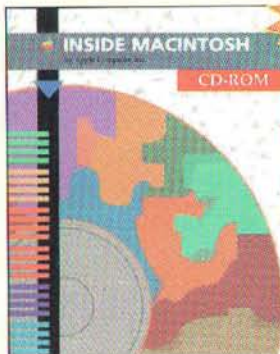


Inside Macintosh: CD-ROM

by Apple Computer, Inc.

More than 25 volumes in electronic form. Includes: QuickDraw™ GX Library, Macintosh Human Interface Guidelines, PowerPC System Software, Macintosh Toolbox Essentials and More Macintosh Toolbox, QuickTime and QuickTime Components. Access over 16,000 pages of information with Hypertext linking and extensive cross referencing.

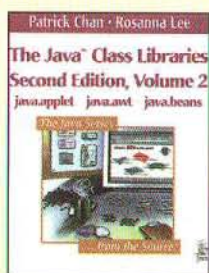
(BIMCD) Our Price **\$89**



HOT PRODUCT

Order Toll-free
800-MACDEV-1
(800-622-3381)

New Books!

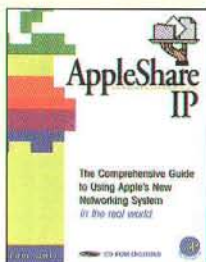


The Java Class Libraries, Second Edition, Volume 2 by Chan Lee

This book is intended as a reference rather than a tutorial. Its format is similar to a dictionary's in that it is designed to optimize the time it takes for you to look up information on a class or class member. The

classes in this book are ordered alphabetically without regard to package name. This makes looking up a class as straightforward as looking up a word in a dictionary. Each class is described in its own chapter. Each chapter contains a picture of the class hierarchy, a class description, a class example, a member summary, and descriptions for every member in the class.

(BJCLASS) Our Price **\$44**



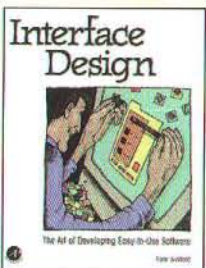
AppleShare IP: By Tom Dell

This book is for administrators either using or planning to use Apple's new networking system. The book discusses choosing servers, configuration, access privileges, the AppleShare file server, print service, e-mail, web services, Apple Search, MacDNS and

interfacing third party products.

- Discusses choosing a server and the AppleShare file service
- Cover AppleShare IP support for PCs
- Addresses AppleShare IP e-mail, web and print services
- Includes hands-on exercise and one CD-ROM for Macintosh with and AppleShare IP Administrator's Toolkit

(BSHARIP) Our Price **\$35**



Interface Design: By Peter Bickford

This book is a practical guide for designing software with users in mind. It offers an on-the-job view of what it takes to create great products, offering practical tips and advice instead of forcing the reader to extrapolate from abstract psychological theory. Interface

Design targets a wide range of design issues, from taming the incomprehensible interfaces of database systems and the Internet, to using sound and animation effectively in multimedia. Throughout the book, the author offers techniques for controlling the growing complexity of computer software, and makes an impassioned case for intelligent design based on the real need of users.

(BINTER) Our Price **\$31**

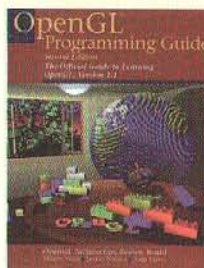


Be Developer's Guide by The Be Development Team

The Be Developer's Guide is the official programmer's reference manual for the BeOS, a revolutionary new operating system built around multimedia, threading, and multiprocessing. Essential reading for anyone who wants to design runnable applications

for the BeOS, this book describes and explains how to use all the development kits, providing multimedia developers access to the internals of the first new operating system in years.

(BBDEV) Our Price **\$45**



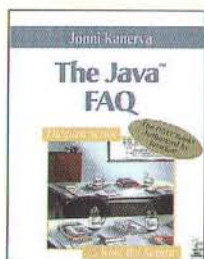
The OpenGL Programming Guide by Mason Woo, Jackieneider and Tom Davis

- Coverage of the new features of OpenGL, Version 1.1, including all texturing changes, vertex arrays, polygon offset, and RGBA logical operations

- The incorporation of the OpenGL Utility Toolkit, GLUT, in all programming examples an overview of the OpenGL rendering pipeline and state machine
- Enhanced coverage of polygon tessellation, quadric surfaces, pixel operations, and error handling

- More performance tips
- A greatly expanded index

(BOPGL) Our Price **\$39**



The Java FAQ by Jonni Kanerva

Java FAQ provides an insider's view of the Java™ technology by posing and answering the most important, frequently asked questions about the Java programming language, Java applets, and Java stand-alone applications. The Java FAQ is unique in that it

draws from the tens of thousands of questions sent to <java@java.sun.com> and provides authoritative answers direct from the creators of the Java programming language at JavaSoft.

(BJFAQ) Our Price **\$25**

Order Toll-free
800-MACDEV-1
(800-622-3381)

APPLE ENTERPRISE SOFTWARE

Getting Started With WebObjects

by Apple Enterprise Software

If you're a first time user, start here to learn the basics of how to create and run WebObjects applications.

(BGSWO) Our Price **\$14**

WebObjects Developer's Guide

by Apple Enterprise Software

A guide to building and understanding WebObjects applications. Takes a close look at the WebObjects scripting language. Additional sections explain the WebObjects architecture and tells you how to integrate your code into the request-response loop, create reusable components, create client-side components, take advantage of powerful Foundation Framework features, and more. Filled with example code. For all WebObjects programmers.

(BWODG) Our Price **\$16**

D'OLE Developer's Guide

by Apple Enterprise Software

Distributed OLE is available today, and this book shows you how to use it. Using real-world examples, the book steps you through the process of making your OPENSTEP objects available on Windows through OLE.

(BDOLEDG) Our Price **\$22**

Discovering OPENSTEP, Mach

by Apple Enterprise Software

Introduces programmers to NeXT's OPENSTEP 4.0 Developer product by guiding them through the creation of three applications of increasing complexity. The tutorials demonstrate and explain programming techniques, Objective-C fundamentals, common APIs, and usage of the development tools. Along the way they present summaries of important concepts and paradigms. The book also includes a chapter directing readers to programming resources, further information, and services such as training and support. An appendix offers a concise discussion of object-oriented programming.

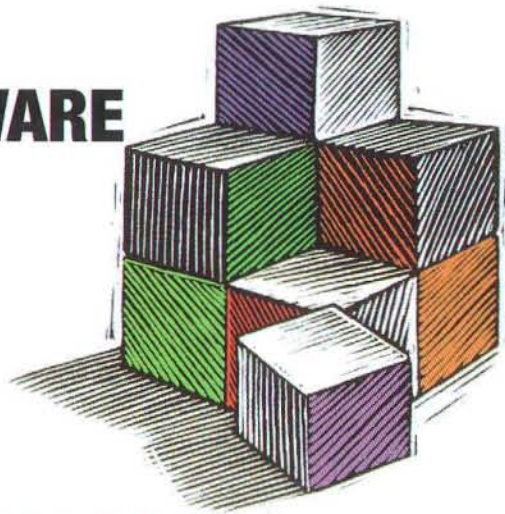
(BDOSTPM) Our Price **\$15**

Discovering OPENSTEP, Windows

by Apple Enterprise Software

Discovering OPENSTEP provides an introduction to OPENSTEP programming on Windows NT. It guides the reader through the creation of three applications of increasing complexity. Along the way, it explains concepts and illustrates aspects of Objective-C, OPENSTEP classes, the development environment, and programming techniques. A short appendix offers a summary of object-oriented programming.

(BDOSTEPW) Our Price **\$16**



Object-Oriented

Programming and Objective C

by Apple Enterprise Software.

An introduction to the principles of object-oriented programming in OPENSTEP and the official description of the Objective-C language. Objective-C is easy to learn and use because it adds very little syntax to the C programming language. Its dynamic nature allows you to accomplish things not possible in most other object-oriented languages. For any OPENSTEP programmer.

(BOOPOC) Our Price **\$24**

Working w/ Interface Builder (for EOF)

by Apple Enterprise Software

A hands-on, award-winning book designed to help you get your job done with the updated Interface Builder, released with NEXTSTEP 3.3 and the Enterprise Objects Framework 1.1. For any programmer using Interface Builder to design objects that truly work in NEXTSTEP.

(BWIB) Our Price **\$24**

Using EOF 2.1 w/ OPENSTEP (Mach & Windows)

by Apple Enterprise Software

Using Enterprise Objects Framework with OPENSTEP describes how to create an Enterprise Objects Framework application on OPENSTEP. It includes a tutorial and a chapter on creating a user interface for an OPENSTEP Enterprise Objects Framework application.

(BUEOFO) Our Price **\$14**

EOF Developer's Guide for EOF 2.1 (Mach & Windows)

by Apple Enterprise Software

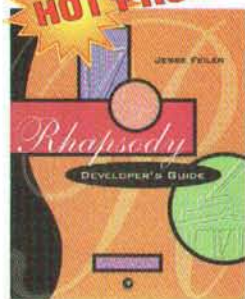
The Enterprise Objects Framework Developer's Guide describes how to develop database applications using the Enterprise Objects Framework tools and classes. It includes an architectural overview of the product, and descriptions of programming tips and techniques. An appendix offers a summary of Entity-Relationship Modeling.

(BEOFDG) Our Price **\$24**

EOF Developer's Guide for EOF 2.0 (BEOFDG20) Our Price **\$24**

EOF Developer's Guide for EOF 1.x (BEOFDG1X) Our Price **\$24**

HOT PRODUCT



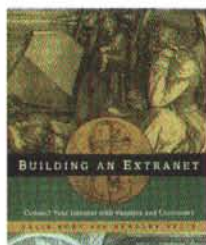
Rhapsody Developer's Guide

by Jesse Feiler

Covers the basic architectural principles of Rhapsody: the Mach microkernel, object-oriented programming, and the elements of a modern OS such as preemptive multitasking, protected memory, and symmetric multiprocessing.

Also shows ways of getting to this new environment — Objective C, conversion tools, and the integration of Java — to develop Rhapsody products. Paperback, 450 pages.

(BRDG) Our Price **\$35**



Building An Extranet: Connect Your Intranet With Vendors And Customers

by Julie Bort and Bradley Felix

Building an Extranet will help companies use their Intranet capabilities to supply information to selected customers and vendors through a

secure Extranet. This book provides complete information and working details for building the network behind Intranets and Extranets, designing the applications, and getting everything up and running.

(BBLDXT) Our Price **\$26.99**



Debugging Macintosh Software

with MacsBug

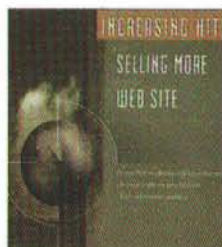
by Konstantin Othmer and Jim Straus

MacsBug, from Apple Computer, Inc., is the leading debugging software program for the Macintosh. This book/disk package is an all-in-one kit



for using MacsBug. Chapter 1 introduces MacsBug and describes the contents of the rest of the book. Chapter 2 describes how to install MacsBug and enough low level details about the Macintosh so that you can use MacsBug. Includes MacsBug 6.2 on disk.

(BDMSWM) Our Price **\$31**



Increasing Hits and Selling More on your Web Site

by Greg Helmstetter

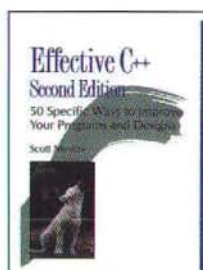
Written especially for entrepreneurs, corporate marketing managers, small business owners, and consultants, this valuable guide gives you rare tips and tricks

you need to know to make your site a commercial success.

(BIHSMWS) Our Price **\$22.45**

Check out our Web site!

• Full product descriptions • Hundreds of more products
<http://www.devdepot.com>



Effective C++, Second Edition:

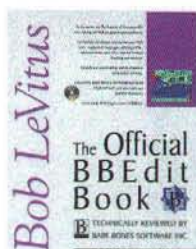
50 Specific Ways to Improve Your Programs and Designs

by Scott Meyers

Effective C++, 2nd Edition includes: Expert guidance on object-oriented design, class design, and the proper use of inheritance

- An examination of the standard C++ library, including how the Standard Template Library and classes like string and vector affect the structure of well-written programs
- Discussions of late-breaking language features like in-class constant initializations, namespaces, and member templates
- Wisdom usually possessed by only the most experienced developers
- Effective C++ continues to be essential reading for every developer working with C++.

(BEFFC) Our Price **\$34**



The Official BBEdit Book

by Bob LeVitus and Natanya Pitts

The Official BBEdit Book makes it easy for today's Webmasters to speed their own interactive development projects using this powerful editing environment. Bare Bones incorporates features like floating palettes, HTML support, and syntax coloring to

enhance an already extensive feature set, making BBEdit a leading Web authoring tool. Super-fast text processing, easy scripting, wide extensibility, strong GREP-style search-and-replace capabilities, and support for 13 languages make it easy to see why so many Web developers use BBEdit as their primary authoring tool.

(BOFBB) Our Price **\$35**



Practical Object-Oriented Development in C++ and Java

by Cay S. Horstmann

This book offers advice on real-world ways to use these powerful programming languages and techniques. Using the Unified Modeling Language (UML) methodology, expert

Cay S. Horstmann gives you clear, concise explanations of object-oriented design, C++, and Java in a way that makes these potentially daunting operations more accessible than they've ever been before.

(BPOOD) Our Price **\$31**



DeBabelizer

by Lise Despres and Paul Vachier

DeBabelizer: The Authorized Edition is the official guide for Web designers, multimedia creators, artists and production specialists who want to take advantage of this powerful tool.

- Create graphics and images for the Web that download fast and look amazing
- Optimize graphics for CD-ROM, video, and animation
- Optimize and manage colors using the SuperPalette(TM)
- Discover DeBabelizer tips and advice from industry experts
- Master basic manipulation techniques, including rotation, scaling, cropping, and text overlay
- Explore key production techniques in all areas of graphics processing

(BDEBTAE) Our Price **\$39**



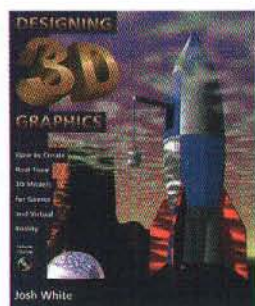
JavaScript & Netscape Wizardry

by Dan Shafer

The perfect book to show you how to turn Netscape into your own personal, customized operating system.

Provides the inside tips and techniques for making your Web pages much more attractive. Shows you how to use all of the key features of the JavaScript language, including objects, methods, properties, events, and much more. Includes CD-ROM with numerous interactive scripts written in JavaScript you can add to your Web pages today. A complete set of the best Java applets. Useful plug-ins designed to supercharge Netscape and resources to help JavaScript programmers.

(BJNWIZ) Our Price **\$31**



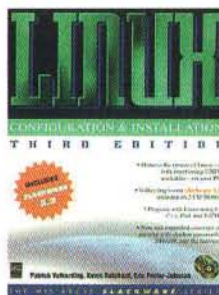
Designing 3D Graphics

by Josh White

In this powerful book/CD-ROM package, top computer graphics artist Josh White tells you everything you need to know to create sophisticated real-time 3D graphics for computer games and virtual reality. This book contains the in-

depth knowledge of software tools and hands-on modeling techniques that Josh White has learned while creating artwork for over 20 commercial games, including Descent, Zone Raiders, Locus, Legoland, and others.

(BD3DG) Our Price **\$35**



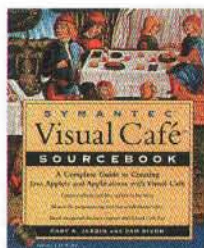
Linux Configuration & Installation, 3rd Edition

by Patrick Volkerking, Kevin Reichard, and Eric F. Johnson

Linux, the leading UNIX variant, has garnered loads of attention within the UNIX community.

The amazing thing about Linux is that you don't need a workstation to run it. Linux Configuration & Installation, Second Edition lets you run Linux today. Program with Linux using C, C++, Perl, and Tcl/Tk. The 2 CD-ROM pack offers one of the most popular Linux distributions, Slackware 96, and comes directly from Patrick Volkerking, the creator of Slackware.

(BLC12) Our Price **\$35**

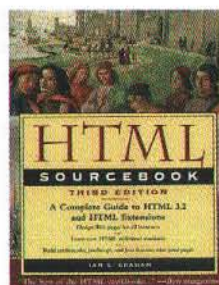


Symantec Visual Cafe Sourcebook

by Cary A. Jardin and Pam Dixon

Symantec Visual Cafe, the first visual Java development tool that gives programmers a sophisticated set of tools. This book teaches programmers how to use Symantec Visual Cafe to create Java applets. It provides a thorough introduction to the language and gives advanced Java programmers information on how to use Visual Cafe to create their own Java development tools.

(BSYMSOUR) Our Price **\$35**



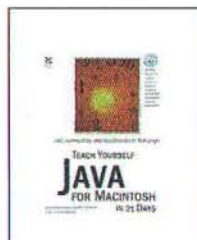
HTML Sourcebook, 3rd Edition

by Ian S. Graham

Critics everywhere agree, HTML Sourcebook is the best guide to HTML for Web professionals. That's because no other book makes it so easy for you to quickly master all the commands, tools, and expert

techniques you need to create cutting-edge Web page documents. Completely revised and expanded by nearly 50 percent, this new third edition of the best-selling guide to HTML gives you the complete lowdown on all the changes and enhancements to the HTML, HTTP, and URL standards.

(BHTMLS) Our Price **\$26.95**



Teach Yourself Java for Macintosh in 21 Days

by Laura Lemay and Charles L. Perkins with Timothy Webster

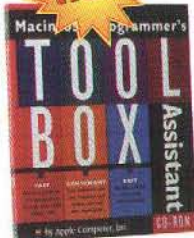
Add interactivity and multimedia to Web pages! A step-by-step guide to make your Website come alive. Learn the basics of programming

Java applets and the concepts behind the Java language. Includes CD-ROM with a limited version of Roaster, the first commercial, integrated applet development environment for Java for the Macintosh!

(BJAVAMAC) Our Price **\$36**



HOT PRODUCT



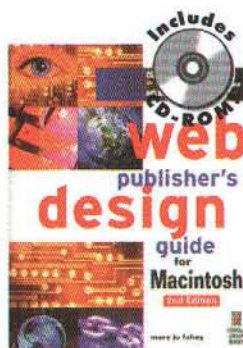
Programmer's Toolbox Assistant CD-ROM

Instant electronic access to Inside Macintosh essentials.
by Apple Computer

Get quick access to reference pages for over 4,000 Toolbox calls in your system software from their development environment. Essential

information for Macintosh software developers. Hypertext links allow programmers to view related topics easily. The ultimate electronic reference tool for Macintosh programmers.

(STBASST) Our Price **\$89**



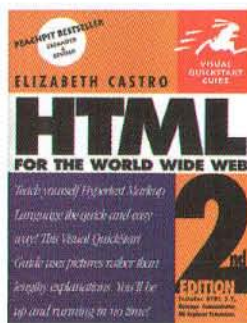
Web Publisher's Design Guide for Macintosh, 2nd Edition

by Mary Jo Fahey

This is the only book that takes you step-by-step through real projects designed by talented new media artists. Internet design experts share their design secrets and art files (look for art

files on the companion CD-ROM by artist's last name). This expanded new edition includes Photoshop, Illustrator and DeBabelizer tricks from the first edition plus countless new ways to liven up your Web pages with 3D graphics, sound, movies, and much more.

(BWPDG2) Our Price **\$35**

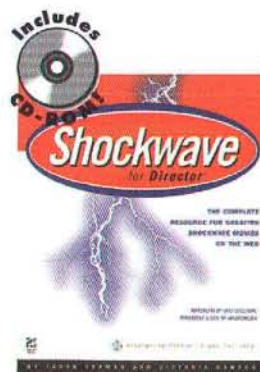


HTML For The World Wide Web, 2nd Edition

by Elizabeth Castro

Teach yourself Hypertext Markup Language the quick and easy way! This Visual QuickStart Guide uses pictures rather than lengthy explanations. You'll be up and running in no time. If you need to learn HTML fast — this book is for you.

(BHTMLW2) Our Price **\$16.15**



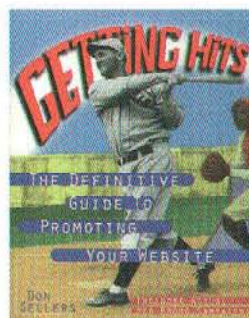
Macromedia Shockwave for Director

by Jason Yeaman and Victoria Dawson

The complete resource for creating Shockwave movies on the Web. This hands-on reference makes it easy to create Shockwave movies and put them on the Web. Expert tips from the creators of Macromedia's first

Shockwave movies, together with detailed examples and instruction, provide everything you need to get started. Includes CD-ROM.

(BMSFD) Our Price **\$27**



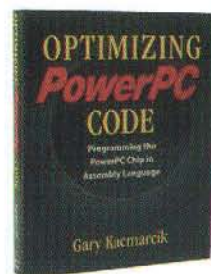
Getting Hits—The Definitive Guide To Promoting Your Website

by Don Sellers

Getting Hits explains in easy-to-understand language the underlying concepts behind the art of Web site promotion. Just a few of the topics you'll learn include: using search

engines with URL's; finding related Internet groups or lists; understanding the nuances of click throughs and ad rates; and distributing press releases to key Internet contacts. With this book, you'll go beyond the conceptual and actually follow real-world tested promotional campaign strategies. Bring the world to your Web site!

(BGHITS) Our Price **\$17.95**

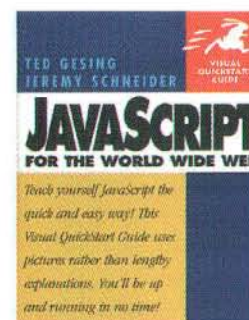


Optimizing PowerPC Code: Programming the PowerPC in Assembly Language

by Gary Kacmarcik

Take full advantage of the potential of the PowerPC by mastering the Assembly Language techniques. Learn to produce faster more robust software!

(BOPTPC) Our Price **\$35**



JavaScript For The World Wide Web

by Ted Gesing and Jeremy Schneider

This book takes an easy, visual approach to teaching JavaScript, where pictures guide you through the software and show you what to do.

Works like a reference book, you look up what you need and then get straight

to work. No long winding passages, concise, straightforward commentary explains what you need to know.

(BJWWW) Our Price **\$16.15**

WebMaster in a Nutshell, Deluxe Edition

by O'Reilly & Associates, Inc.

Cross-platform, completely portable, and lightning fast, the CD-ROM is an invaluable addition to the webmaster's toolbox. The CD-ROM contains the Web Developer's Library — the full text of the latest editions of five popular O'Reilly titles: "HTML: The Definitive Guide, 2nd Edition"; "JavaScript: The Definitive Guide, 2nd Edition"; "CGI Programming on the World Wide Web"; "Programming Perl, 2nd Edition"; and "WebMaster in a Nutshell." The Deluxe Edition also includes a printed copy of "WebMaster in a Nutshell," the all-inclusive quick reference that belongs next to every webmaster's terminal. Includes CD-ROM & 356 page book.

Requirements: The CD-ROM is readable on all platforms, but requires a web browser that supports HTML 3.2, Java, and JavaScript.

(BWMNUTD) Our Price **\$62**

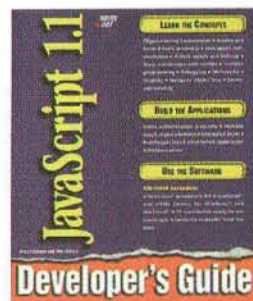


CodeWarrior Software Development Using PowerPlant

by Jan L. Harrington

C++ programmers will learn to develop object-oriented software applications for the Mac and Power Mac using the PowerPlant environment and the classes that support it. Covers CodeWarrior 8. Included CD-ROM contains source code for all the programming examples in the book and Metrowerks CodeWarrior Lite.

(BCWSWDEV) Our Price **\$31**



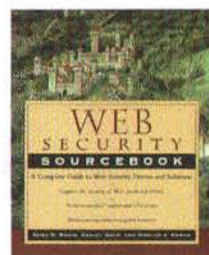
JavaScript 1.1 Developer's Guide

by Arman Danesh and Wes Tatters

Written by developers for developers. An advanced guide to creating professional Web applications with JavaScript 1.1 as deployed in Netscape

Navigator 3.0, Microsoft Internet Explorer 3.0, and LiveWire. Includes CD-ROM with Sun's Java Developer's Kit, JavaScript and HTML Editors for Windows and Macintosh, 20 contributed ready-to-run JavaScripts and JavaScript examples from the book.

(BJSDG) Our Price **\$44**



Web Security Sourcebook

by Aviel D. Rubin, Daniel Geer and Marcus J. Ranum

Technical tools and techniques for building secure web sites and applications

This book shows web masters, web managers, and web designers the hands on programming techniques necessary to build secure web sites. Readers will learn how to secure the server, use firewalls and cryptography, write secure Java applets and CGI scripts and more. Companion Web Site includes source code examples plus updates on the latest security threats and techniques.

(BWEBSE) Our Price **\$26.99**



Wireless For The Newton

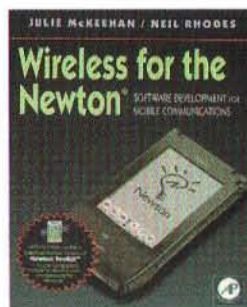
by Julie McKeehan and Neil Rhodes

A book that picks up where Programming for the Newton left off, teaching the reader how to develop Newton software on the Macintosh. The enclosed floppy disk provides a sample application, as well as a fully functional

demonstration version of Newton Toolkit.

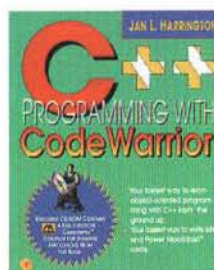
- Learn to develop Newton software on the Macintosh
- Hands-on Newton environment training with sample code
- Includes disk with sample source code for a Newton application, as well as demonstration NTK – the complete development environment for the Newton

(BWIRELESS) Our Price **\$31**



Check out our Web site!

• Full product descriptions • Hundreds of more products
<http://www.devdepot.com>

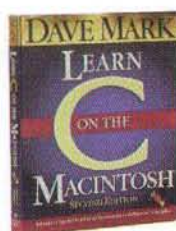


C++ Programming with CodeWarrior

by Jan L. Harrington

Beginning OOP for the Macintosh and Power Macintosh and Mac OS compatibles. Learn object-oriented programming techniques using C++ as the example language and Metrowerks and CodeWarrior as the example compiler. Enclosed CD contains example code from the book and a full-function Metrowerks CodeWarrior.

(BCPPCW) Our Price **\$33**

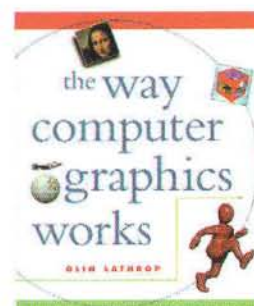


Learn C on The Macintosh, Second Edition

by Dave Mark

New revised edition! Easy-to-understand – everything you need to start programming. Updated and enhanced exercises that lead you step by step. You'll learn function, variables, point datatypes, data structures, file input and output and more! Includes CD-ROM with Metrowerks CodeWarrior™ Lite.

(BLEARNC2) Our Price **\$33**



The Way Computer Graphics Works

by Olin Lathrop

A complete guide to mastering computer graphic basics. It is written in a frank, down-to-earth style covering everything from how computer graphics are different from fine art and photographs, to modeling, pixels, and the principles of animation.

All of this is done without resorting to mind-numbing equations and impenetrable technical jargon.

(BWCGW) Our Price **\$29.65**

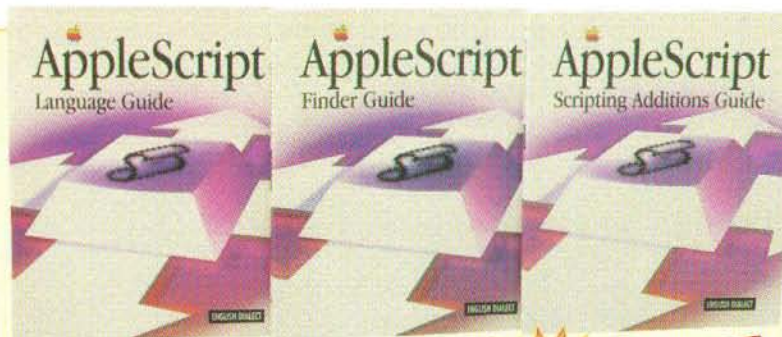
Inside PowerPlant

by Metrowerks

Create PowerPlant applications using the CodeWarrior IDE and PowerPlant Constructor. Full descriptions of major PowerPlant classes and resources. Included are the PowerPlant Constructor Manual, including View, TextTraits and Custom Types editing, and PowerPlant Library Reference, covering all classes and functions in PowerPlant.

(BINSPP) Our Price **\$34**

SEE RELATED CATEGORY: Dev. Environment



AppleScript Language Guide

by Apple Computer, Inc.

A complete reference for anyone using AppleScript to modify existing scripts or to write new ones. Contains useful information for programmers who are working on scriptable applications or complex scripts. Features detailed definitions of AppleScript terminology and syntax in the following categories: Value classes, commands, objects and references to objects, expressions, control statements, handlers, and script objects. Includes many sample scripts, discusses advanced topics such as writing command handlers for script applications, the scope of script variables and properties declared at different levels in a script, and inheritance and delegation among script objects.

(BALG) Our Price **\$26.95**

SEE RELATED CATEGORY: Scripting

HOT PRODUCT

HOT PRODUCT



AppleScript Applications:

Building Applications with FaceSpan and AppleScript

by John Schettino Affiliation & Liz O'Hara

Build complete AppleScript applications using FaceSpan, a user interface development tool that makes AppleScript applications truly "Mac-Like". Uses a step-by-step approach to demonstrate techniques for building applications through illustrations and samples. Provides Graphical User Interface (GUI) design tips and practical approaches for implementation. Contains one CD-Rom with AppleScript 1.1, a demonstrations version of FaceSpan 2.1, source code for all example applications numerous AppleScript shareware and demonstrations programs. Contains a section on debugging AppleScript applications using FaceSpan.

(BAPSCAP) Our Price **\$31**



Special Edition Using CGI, 2nd Edition

by Jeffrey Dwight, Michael Erwin and Robert Niles

This complete reference provides professional Web developers and advanced personal users with the latest information

on using CGI (Common Gateway Interface) to interact with databases.

- Explains client and server uses of CGI
- Provides extensive coverage of live audio and video feeds, user chat and interaction, and CGI security
- Features separate chapters devoted to language-specific tips, tricks, and traps
- CD ROM is loaded with the HTML and CGI sample code from the book
- Includes applications for guest books, mail and new gateways, browser identification, access restriction, and shopping carts

(BSEUCGI) Our Price **\$44**



Java in a Nutshell, 2nd Edition

by David Flanagan

A detailed overview of all of the new features in Java 1.1, both on a package-by-package basis and in terms of overall functionality. A comprehensive tutorial on "inner classes" that explains how to use all of the new types

of inner classes: static member classes, member classes, local classes, and anonymous classes. Practical, real-world example programs that demonstrate the new features in Java 1.1, including object serialization, the new AWT event handling model, internationalization, and a sample Java Bean.

(BJNUT2) Our Price **\$17.95**

AppleScript Finder Guide, English Dialect

by Apple Computer, Inc.

Provides definitions for Finder object classes and commands. Write, record, or run scripts that trigger the same desktop actions that you trigger using the keyboard and mouse.

(BAFG) Our Price **\$17.95**

SEE RELATED CATEGORY: Scripting



JavaScript for the Macintosh

by Matt Shobe and Tim Ritchey

Allows non-programmers to take advantage of the power of Netscape Navigator. Expand the capabilities of your Web page, without having to understand C or C++. CD-ROM contains "Wizlets" that allows you to easily

create your own JavaScripts. Takes you step-by-step through programming cross-platform JavaScripts. Details how to create JavaScripts for JavaScript-aware Web browsers.

(BJAVASCRIPTJ) Our Price **\$40**



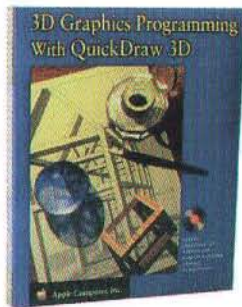
Inside CodeWarrior Professional

by Metrowerks

Includes CodeWarrior IDE User's Guide. This is the printed version of the documentation provided on the CD. Covers CodeWarrior Professional Release, the debugger and associated tools.

(BINSWP) Our Price **\$34**

SEE RELATED CATEGORY: Dev. Environment



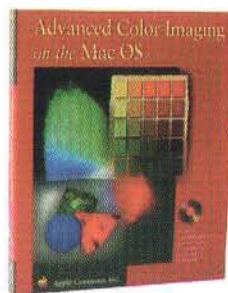
3D Graphics Programming Using QuickDraw 3D

by Apple Computer, Inc.

Incorporate spectacular 3D graphics into your applications. Explore QuickDraw 3D, a revolutionary graphics extension to the Mac OS for Power Macintoshes. CD contains the complete QuickDraw 3D

system itself and a complete database of the QuickDraw 3D API, allowing you instant access to the hundreds of graphics calls via a fast viewing engine. Book/CD-ROM, 640 pages.

(B3DGRAP) Our Price **\$35**



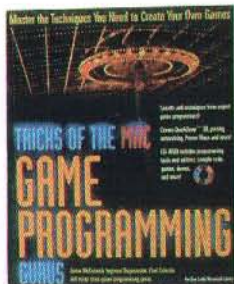
Advanced Color Imaging on the Mac OS

by Apple Computer, Inc.

Enhance your software's color capabilities with step-by-step instructions. Augment the color support supplied with QuickDraw, and QuickDraw GX. Use the Palette Manager to get the best colors on limited displays. Match

colors between screens and input/output devices (scanners & printers). CD includes a complete reference information in both QuickView and Acrobat formats. Plus, a sample application demonstrating ColorSync programming techniques.

(BADVCI) Our Price **\$33**



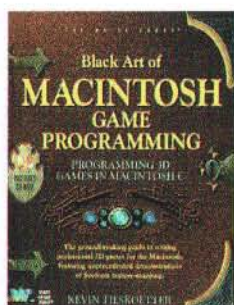
Tricks of the Mac Game Programming Gurus

by McCormack, Ragnemalm, Celestin, et al.

For beginning to expert game programmers. Complete overview of all the necessary components of game programming on the Macintosh. Packed

with valuable tools, utilities, sample code, CodeWarrior™ Lite and game demos. QuickDraw 3D and Power Mac optimization and inside info on how Glypha III was created. Hundreds of tried-and-true tricks, tips, and insider secrets from well-known Mac game programming experts.

(BTRICKS) Our Price **\$45**



Black Art of Macintosh Game Programming

by Kevin Tieskoetter

Develop your own 3D games in C on the Mac. Includes CD with project files for both Symantec C and Code Warrior. Create freeform texture-mapped games and polygon graphics. Control dynamic source code — all compatible as native to the Power Mac. Write directly to the screen, bypassing QuickDraw.

(BBLACK) Our Price **\$35**



Here are more products. For full product descriptions please see our Web site, or feel free to call, fax, or E-mail us. **Our prices on books are at least 10% off list price.**

PRODUCT	CODE	PRICE
Development Environments		
AppleScript Applications: Building Applications w/FaceSpan.....	BAPSCAP	\$31.00
C++ Programming.....	BCPPMACAP	31.00
C/C++ SDK User's Guide.....	BCPPUSER	29.00
CodeWarrior Inside PowerPlant.....	BINSPP	34.00
Cyberdog Programmers Kit.....	BCYBERDOG	34.00
Dan Shafer Presents the Power of Prograph CPX.....	BDANPRO	19.95
Inside CodeWarrior Book.....	BINSCW	34.00
Instant CORBA.....	BIC	17.99
Last Resort Programmers Edition.....	BLSTRISRT	74.00
Macintosh C Programming Primer Volume I.....	BCPRIM1	24.25
Macintosh C Programming Primer Volume II.....	BCPRIM2	24.25
Macintosh Pascal Programming Primer Volume I.....	BPASCPII	24.25
Mastering the Think Class Library.....	BMASTERTCL	26.95
Mastering the Toolbox Using THINK C.....	BCPRIM2	24.25
Objective-C - Object-Oriented Programming.....	BOBJCOOPT	34.00
Presenting Magic Cap.....	BPRESMAGIC	15.25
Real World Apple Guide.....	BREALWLD	35.00
Symantec C++ Programming.....	BSYMCPP	39.00
Taligent's Guide to Designing Programs.....	BTALIGENT	17.55

PRODUCT	CODE	PRICE
The Power of Prograph CPX.....	BDANPRO	19.95
Visual Programming with Prograph CPX.....	BVISPRO	30.00
Wireless For The Newton.....	BWIRELESS	31.00

Hardware

Apple CD-ROM Handbook.....	BCDHAND	14.36
Designing Cards & Drivers for the Macintosh.....	BCARD	26.96
LaserWriter Reference.....	BLASERREF	17.96
PCI System Architecture 3rd Edition.....	BPCISYS	31.00
PowerPC System Architecture.....	BPCCARCH	31.00

Internet Related

1994 Internet White Pages.....	B94WHITE	26.95
Active Java.....	BACT.JAVA	23.36
America Online for Dummies.....	BAOLDUM	17.95
CGI By Example.....	BCGIBE	31.00
Building & Maintaining an Intranet w/ the Macintosh.....	BBAMJ	45.00
Claris Home Page Companion.....	BCHP	26.95
Computer Privacy Handbook.....	BPRIV	22.45

Developing CGI Applications with Perl	BDCGIA	26.95
E-Mail Essentials	BEMAIL	22.45
Elements of E-Mail Style	BEMAIL	13.45
Hooked on Java	BHJAVA	26.95
Instant Internet Guide	BINSTANT	13.45
Internet Book	BHNET	22.50
Internet for Dummies 2nd Edition	BNETDUM2	17.99
Internet for Dummies Quick Reference	BDUMOCK	8.05
Internet for Macs for Dummies	BNETDUM	17.95
Internet for Macs for Dummies Bestseller Edition	BIFMFOBE	35.00
Internet Power Tools	BPWRTOL	36.00
Internet Publishing with Adobe Acrobat	BIPWAA	36.00
Internet, The, Deluxe Edition	BNETDELUX	31.00
Intranet Web Dev.: Enterprise Alternatives to Client/Server	BINTWD	44.00
Java Essentials for C/C++ Programmers	BJAVASSEN	17.95
Java in a Nutshell	BJAVANUT	13.45
Java Language API SuperBible	BJLAS	53.00
Java Programming with CORBA	BJPWC	26.99
JavaScript Cookbook	BJSCB	44.00
JavaScript for Macintosh	BJAVASCRIPT	40.00
Learn HTML on the Macintosh	BLHTML	26.95
Learn Java on the Macintosh	BLJAVA	31.00
Mastering Netscape 2.0 for Macintosh, Second Edition	BMASNET2	36.00
Measuring the Impact of Your Web Site	BMIYWS	26.95
More Internet for Dummies Starter Kit	BDUMNET	17.95
Mosaic for Dummies	BMOSDUM	17.99
Net Chat	BNETCHAT	17.00
NetObjects Fusion Handbook	BNETOFH	45.00
Perl Quick Reference	BPERLREF	17.99
Planning and Managing Websites	BPLANWEB	35.00
Protect Your Privacy on the Internet	BPYP	26.99
Providing Internet Services via the MacOS	BPROVNET	31.00
Publish It on the Web	BWEBPUB	31.00
TCP/IP Vol 1-Vol 2 Bundle	BTCP12BNDL	99.00
Teach Yourself Java in 21 days	BJAVAMAC	36.00
The Internet Marketing Plan	BTIMP	35.00
The Internet Strategic Plan	BTISP	22.45
The Java FAQ	BJFAQ	25.00
The Web Navigator	BTWN	22.45
Underground Guide to Telecommuting	BUNDER	22.45
Using Lotus Notes as an Intranet	BULN	40.00
Web Graphics Tools and Techniques	BWGTT	35.00
Web Head Mac Guide	BWEBHEAD	22.45
Web Marketing Cookbook	BWMCB	35.00
Web Page Scripting Techniques	BWEBPST	45.00
Web Publishing with Adobe Acrobat and PDF	BWPWAA	35.95
Web Weaving	BWWEAV	22.45
Webmaster Macintosh	BWEBMAS	26.95

Scripting and Solutions

AppleScript Applications: Building Apps with FaceSpan	BAPSCAP	31.00
AppleScript Scripting Additions Guide	BSCRADD	17.05
Applied Macintosh Scripting	BAPPLIED	31.00
Complete HyperCard 2.2 Handbook	BHYPCRD2	31.00
Complete HyperTalk 2.2:	BHYPCRD2	31.00
Danny Goodman's Apple Guide Starter Kit	BGAGSK	31.00
HyperCard Stack Design	BHYPSA	19.95
JavaScript for Macintosh	BJAVASCRIPT	40.00
Perl Quick Reference	BPERLREF	17.99
Real World Apple Guide	BREALWLD	35.00

Technical Reference

Active Java	BACTJAVA	23.36
Apple CD-ROM Handbook	BCDHAND	14.36
AppleShare IP	BSHARP	35.00
Art of Human Interface Design	BAHD	29.65
Be Developer's Guide	BBDEV	45.00
Black Art of Mac Game Programming	BLACK	35.00
C++ for Dummies	BCPPDUM	22.46
C for Dummies Vol. 1	BCDUM	17.95
Developing Object Oriented Software for the Macintosh	BDEVOB	26.06
Extending the Mac Toolbox	BETMT	22.46
Foundations of Macintosh Programming	BFOUND	35.00
Fragment of Your Imagination	BFRAG	35.00
Guide to Macintosh Software Localization	BLOCALIZ	24.26
Guide to Macintosh System 7.5	BSYS7.5	22.50
Inside AppleTalk	BAPTALK	31.00
Inside the Macintosh Communications Toolbox	BCOMM	22.45

LaserWriter Reference	BLASERREF	17.96
Learn C++ on the Macintosh	BLRNCPP	35.00
Learn C on the Macintosh, 1st Edition	BLRNC1	31.00
Learn C on the Macintosh, 2nd Edition	BLRNC2	33.00
Mac Programming for Dummies	BMACDUM	17.95
Macintosh C Programmer Primer Volume 1	BCPRIM1	24.25
Macintosh C Programmer Primer Volume 2	BCPRIM2	24.25
Macintosh OLE2 Prog. Reference Working with Objects	BOLE2	40.00
Macintosh Pascal Programming Primer Volume I	BPASCPRI	24.25
Macintosh Programming Secrets 2nd edition	BPSECRET	28.76
Macintosh Programming Techniques	BPTECH	31.00
Microsoft Visual J++ 1.1 Sourcebook	BMVJS	35.00
More Mac Programming Techniques	BMORETECH	34.00
Network Frontiers Bundle	BNETFB	59.00
Newton Programming Guide	BNEWTPGUID	40.00
Object Oriented Programming Design	BOOPRODES	20.66
Optimizing PowerPC Code	BOPTPPC	35.00
Perl Quick Reference	BPERLREF	17.99
PostScript Language Reference	BPSLANREF	29.66
Powerbook: Digital Nomad's Guide	BPBTONG	22.46
PowerPC Programmer's Toolkit	BPPCPT	40.00
Programming Introduction to the Macintosh Family	BFAMILY	22.46
Programming for System 7	BSYS7	24.25
Programming with AppleTalk	BPROAT	22.45
QuickTime - Official Guide for Macintosh Users	BQTGUIDE	45.00
Real World Apple Guide	BREALWLD	35.00
ResEdit All Night Diner	BRESIDNE	22.45
ResEdit Complete, 2nd Edition	BRESED2	31.00
ResEdit Reference	BRESEDRF	26.96
Software by Design: Creating User Friendly Software	BDESIGN	26.95
Symantec C++ for the Macintosh: The Basics	BSCFMTB	31.00
Teach Yourself Macintosh C++ in 21 Days	BCPP21D	26.99
Technical Introduction to the Macintosh Family	BTITIME	24.26
The OpenGL Programming Guide	BOPGL	39.00
The Java Class Libraries, Second Edition, Volume 2	BJCLASS	44.00
Tog on Software Design	BTOG	26.95
Writing Localizable Software	BLOCAL	24.25

Miscellaneous

3D Graphics-Tips, Tricks, & Techniques	B3DGTIT	31.00
A Fragment of Your Imagination	BFRAG	35.00
Adobe Premiere for the Macintosh	BPREM	44.00
America Online for Dummies	BAOLDUM	17.95
AppleGuide Complete	BAPLG	35.00
Art of Human Interface Design	BAHD	35.00
CD-ROM Guide to Multimedia Authoring	BCDMULTI	40.00
CompuServe for Dummies	BCSDUM	17.95
Cyberpunk Handbook	BCYPUNK	8.95
Danny Goodman's Apple Guide Starter Kit	BGAGSK	31.00
Danny Goodman's Macintosh Handbook	BGOODHB	26.95
FrameWorks Source Code Disk	MTFWSC	9.95
FrameWorks Magazine Back Issue	MTFWBACK	8.00
Graphic Gems 2	BGEMS2	44.00
Graphic Gems 4	BGEMS4	44.00
Graphic Gems V	BGEMS5	44.00
Infini-D Revealed	BINFREV	40.00
Inside Director 5 with Lingo for Macintosh	BIDSWLM	44.00
Interface Design	BINTER	31.00
Late Night with MacHack	BLATE	26.95
Mac Bathroom Reader	BBATH	11.70
Macromedia Director Lingo Workshop, 2nd Edition	BMDLW2	40.00
Mac Screamer: The Ultimate Macintosh Supercharging Kit	BSCREAM	31.00
Macintosh Crash Course	BCRASH	26.95
MacTech Back Issues	MTBACKISS	10.00
Macworld Ultimate Macintosh Programming Book	BULTMAC	35.00
MADACON '93 CD-ROM	SMADA93	9.95
Multimedia Authoring: Building and Developing Documents	BMMALUTH	31.00
Multimedia Starter Kit for Macintosh	BMMSTART	27.00
Profit from Experience	BPROFIT	22.45
ResEdit Complete, Second Edition	BRESED2	31.00
Sad Macs, Bombs and Disasters	BSADMAC	22.45
Standards For Online Communication	BSFOC	40.00
The Elements of E-Mail Style	BEMAIL	13.45
The Software Developer's & Marketer's Legal Companion	BSDAML	33.00
Tog on Software Design	BTOG	26.95
Tricks of the Mac Game Gurus	BTRICKS	45.00
Zen and the Art of Resource Editing	BZAAORE	27.00

All entries in this index are alphabetized. For your convenience the product names are **bold**, book names are *italicized* and company names are in plain text.

1000 Games for Macintosh.....	19
3D Graphics Programming Using QuickDraw 3D.....	29

—A—

AAA+ Software	6
Absoft Corporation.....	3
Abuse	19
Adamation.....	9
Additional Listings for Development Environments	5
Additional Listings for Games	19
Additional Listings for Internet	13
Additional Listings for Multimedia	18
Additional Listings for Tools, Libraries & Utilities	11
Adianta Inc.....	10
Advanced Color Imaging on the Mac OS.....	29
AG Author	10
Altura Software, Inc.....	5
Apple Enterprise Software.....	23
AppleScript Applications: Building Apps w FaceSpan/AppleScript.....	28
AppleScript Finder Guide, English Dialect.....	28
AppleShare IP.....	22
AppleScript Language Guide.....	28
AppMaker	9
Apprentice 7	7
AudioTrack	18

—B—

B-Tree HELPER 2.2	8
Bare Bones Software.....	8, 10, 12
BBEdit 4.5	8, 12
Be Basics	11
<i>Be Developer's Guide</i>	22
Be, Inc.....	3
Beale Street Group, Inc.	15
BeOs Preview Release 2	3
BeachWare, Inc.....	7, 18, 19
BeatWare.....	11
BeSpecific 3	9
<i>Black Art of Macintosh Game Programming</i>	29
Blueworld Communications	12
Bowers Development.....	9
<i>Building An Extranet: Connect Your Intranet w Vendors/Customers</i>	24
Bungie Software.....	19

—C—

C++ Programming with CodeWarrior.....	27
Captivate 4.6 Essential Graphics Utilities	18
Celestin Company.....	7
CGi Toolkit	13
Classic Arcade	19
Clip VR™	18
CodeBuilder	5, 8
CodeWarrior Discover Programming Editio	3
CodeWarrior for BeOS 3	2
CodeWarrior for PalmPilot Release 4	2
CodeWarrior Latitude	2
CodeWarrior Professional Release 2	3
<i>CodeWarrior Software Development Using PowerPlant</i>	27
Conix Graphics.....	11
CyberStudio	12, 15

—D—

<i>D'OLE Developer's Guide</i>	23
<i>DeBabelizer</i>	25
<i>Debugging Macintosh Software with MacsBug</i>	24

<i>Designing 3D Graphics</i>	25
Digital Technology International.....	14
Digitool, Inc.....	3
<i>Discovering OPENSTEP, Mach</i>	23
<i>Discovering OPENSTEP, Windows</i>	23
DynaMorph 1.5	14

—E—

<i>Effective C++, Second Edition</i>	24
<i>EOF Developer's Guide for EOF 2.1(Mach & Windows)</i>	23
ExposurePro	15

—F—

FaceSpan v3.0	14
Future Basic II	10

—G—

GeekWare™	20
GenieWorks, LLC	6
<i>Getting Hits—The Definitive Guide To Promoting Your Website</i>	26
<i>Getting Started With WebObjects</i>	23
GoLive.....	12, 15
Guide Composer™ 1.2	8

—H—

<i>HTML For The World Wide Web, 2nd Edition</i>	26
<i>HTML Sourcebook, 3rd Edition</i>	25
HyperGuide 1.0	12, 18

—I—

<i>Increasing Hits and Selling More on your Web Site</i>	24
<i>Interface Design</i>	22
<i>Inside CodeWarrior Professional</i>	28
<i>Inside Macintosh: CD-ROM</i>	21
<i>Inside PowerPlant</i>	27

—J—

<i>Java in a Nutshell, 2nd Edition</i>	28
<i>JavaScript 1.1 Developer's Guide</i>	27
<i>JavaScript for the Macintosh</i>	28
<i>JavaScript & Netscape Wizardry</i>	25
<i>JavaScript For The World Wide Web</i>	26
Joy Explorer	6
Joy Developer	6

—K—

Kaidan.....	16-17
KiWi	17
KiWi and KiWi+ Accessories	17
KiWi+ Detent Discs	17
KiWi Flash Hotshoe Level	17
KiWi Landscape Bracket	17
KiWi QuickTilt Leveler	17
KiWi-to-KiWi+ Upgrade	17

—L—

Lakewood Software.....	10, 18
Lasso 2.5	12
Late Night Software Ltd.....	14
<i>Learn C on The Macintosh Second Edition</i>	27
<i>Linux Configuration & Installation, 3rd Edition</i>	25

—M—

MacA&D 6.0	7
Macintosh Common Lisp 4	3

Macromedia Shockwave for Director	26
MacTech® CD-ROM Volumes 1-12	21
MacTech® Magazine	21
MacTech® Mouse Pad	20
Magellan Accessories	16
Magellan QC	16
Magellan QC Detent Wheels	16
Magellan QC Pedestal Set	16
Magreable Software	8
Main Event Software	14
Mainstay	4, 13
Mango Tree Software	14
Marathon Trilogy Box Set	19
Maxum Development	13
Media Cleaner Pro	18
Memory Mine	10
Metrowerks	2, 3, 20
MkLinux: Microkernel Linux for the Power Macintosh	4
Morph	14
MultiWare Multimedia Collection	18
Music Tracks	18
MW Visual SourceSafe Release 5	2
Myth The Fallen Lords	19

—N—

Neon Software	6
NewtCard	15
Music Tracks	18
NetMinder Ethernet	6
Nodester	15
NS BASIC Corporation	5, 15
NS BASIC 3.6 for the Newton	5

—O—

Object-Oriented Programming and Objective C	23
ObjectMaster Professional Edition	5
ObjectSet Mail SDK	8, 12
Offset Spacer	17
Onyx Technology, Inc.	7
OpenGL for the Macintosh	11
Optimizing PowerPC Code: Programming the PowerPC in Assembly Language	26

—P—

PageCharmer 1.0	13
Panimation	15
Pictorius Inc.	13
Pilot Attaché Disk 1	9
Podium Sport	20
Power MachTen 4.0.3	13
Power3D	19
PowerKey Pro Model 200, 600	6
Practical Object-Oriented Development in C++ and Java	24
PreFab Player	14
PreFab Software, Inc.	14
Prime Time Freeware	4
Pro Fortran	3
Programmer's Toolbox Assistant CD-ROM	26

—Q—

QC	7
Quasar Knowledge Systems, Inc.	5
QuickPan Counterweighting Kit	16
QuickPan Detent Wheel	16
QuickPan Magnum	16
QuickPan Magnum Accessories	16
QuickPan QuickTilt Leveler	16
QuickTime VR Authoring Studio, Kaidan Bundle	16

—R—

Rach, Inc.	20
Rhapsody Developer's Guide	24

Royal Software, Inc.	14
Rumpus	13

—S—

Script Debugger	14
Scripter 2.0	14
ScriptGen Pro	8
Seapine Software, Inc.	9
SmallTalkAgents for Macintosh	5
Smartcode Software	8, 12
SoftPolish CD-ROM	10
Sophisticated Circuits	6
Special Edition Using CGI	28
SpotCheck	6
Spotlight	7
Staz Software	10
Step-Up Installer Pack	8
StepUp Software	8
StoneTable 68K/PPC	7
StoneTable Publishing	7
SuperAnalyst	10
SuperPlotPRO	10
SuperSoft	10
Symantec	4
Symantec Visual Cafe Sourcebook	25

—T—

TCP/IP Scripting Addition	14
Teach Yourself Java for Macintosh in 21 Days	25
Techworks	19
Tenon Intersystems	5, 8, 13
Terran Interactive	18
TestTrack-Bug Tracking the Macintosh Way	9
The Java Class Libraries, Second Edition, Volume 2	22
The Java FAQ	22
The Official BBEdit Book	24
The OpenGL Programming Guide	22
The Way Computer Graphics Works	27
Tools Plus libraries + framework	9
Tricks of The Mac Game Programming Gurus	29

—U—

UNI SOFTWARE PLUS	7
Using EOF 2.1 w/ OPENSTEP (Mach & Windows)	23

—V—

VIP-BASIC: Visual Interactive Programming in BASIC	4
VIP-C: Visual Interactive Programming in C	4
Visual Cafe for Java Professional Development Edition	4
Visual Cafe for Java Database Development Edition	4
Visual MacStandardBasic 3.0	7
VOODOO 1.8	7

—W—

Water's Edge Software	9
Web Publisher's Design Guide for Macintosh, 2nd Edition	26
Web Security Sourcebook	27
Web Ware	7
WebMaster in a Nutshell, Deluxe Edition	26
WebObjects Developer's Guide	23
WebTen	12
Widgetizer	15
WindowScript	14
Wireless For The Newton	27
Working w/ Interface Builder (for EOF)	23

—X—

Xplain Corporation	20, 21, 27
--------------------------	------------

—Z—

ZCurve Software	7
-----------------------	---

APS The Drives to Succeed!

Prices in Red are Hot Deals! Order online @ <http://www.apstech.com>

Mac Backup^{as low as} \$349⁹⁵!

APS HyperQIC®

The best value in a high performance tape backup system! The APS HyperQIC uses reliable, industry-standard Travan 4 tape cartridges to backup up to 8GB* of data on one tape.

HyperQIC Starter Kit

(includes three TR-4 cartridges, one drive cleaner kit) part # 106625 \$119.95

*Assumes 2:1 data compression

Upgrading to System 8.1?

According to Apple, before you can take advantage of the improved hard disk management of Mac OS 8.1, you must back up the entire contents of your drive. The APS HyperQIC in the portable, stackable SR 2000 enclosure is the most economical way to do that.

\$599⁹⁵

APS Pro Enclosure

Save on the APS HyperDAT®

- Up to 8GB on a single 120m DAT tape**

- Includes Mac/PC/NT backup software

Now just \$599.95 in the APS Pro enclosure!

**Assumes 2:1 data compression

120M DAT Starter Kit (includes five DAT tapes, one DAT head cleaner, one LabelOnce DAT starter kit, one Micro-Data storage cube) part # 100590 \$99.95

APS JAZ

- Macworld Editors' Choice
- Huge 1GB capacity on removable media

*Does not include cartridge

\$299⁹⁵*

APS Pro Enclosure

APS ECONOMY LINE DRIVES

- Perfect replacement drives
- No expensive extras that you don't need

APS Q 2000 EL	2079MB formatted, 5400 rpm Quantum mechanism	\$199 ⁹⁵
APS Q 3000 EL	3118MB formatted, 5400 rpm Quantum mechanism	\$229 ⁹⁵
APS Q 4000 EL	4136MB formatted, 5400 rpm Quantum mechanism	\$279 ⁹⁵
APS Q 6400 EL	6260MB formatted, 5400 rpm Quantum mechanism	\$359 ⁹⁵

The APS Economy Line of drives is intended as replacements for your current hard disk. They do not include brackets, cables or screws. Please keep your invoice for warranty purposes.

APS Q 4300 W

- Productivity boost for creatives
- Faster transfers with Ultra Wide SCSI

Shown in APS Pro Enclosure

PCI ULTRA WIDE SCSI HOST ADAPTER

ADAPTEC POWERDOMAIN™ 2940UW FOR MAC

With the purchase of any APS Ultra Wide SCSI drive

Mac OS is a registered trademark of Apple Computer, Inc. APS and APS Technologies are registered trademarks of Alliance Peripheral Systems, Inc. Other brand or product names are registered trademarks or trademarks of their respective holders.

- 30-day money-back satisfaction on all APS hard drives and accessories. Your risk is the cost of shipping.
- Monitors and 3rd party products carry warranties from their respective manufacturers.
- SCSI cables sold separately.
- APS warranty is a pass-through of the manufacturer's warranty which will be provided by APS for one year. APS warrants APS branded enclosures, brackets and power supplies for a period of one year from the date of purchase and will, at its discretion, replace or repair products found to be defective according to the specific terms of the warranty accompanying the product.
- Refund orders subject to 20% restocking fee.
- International customers must pay for shipping charges.

- Listed capacities are formatted.
- Actual data compression and tape capacity vary greatly depending on the type of data recorded, other system parameters and environment.
- Prices and specifications are subject to change without notice.
- New install system software appropriate to your machine before using our hard drives.
- Not responsible for typographical errors.
- Some APS Drive unit may contain one or more components that were in prior use, but which have been inspected and repaired or rebuilt as necessary. These components, if any, are covered by the same warranty provisions, enclosures and limitations as the same extent, and on the same terms as otherwise "new" components.
- © 1998, Alliance Peripheral Systems, Inc. All Rights Reserved.

APS
Technologies

APS HIGH-PERFORMANCE ULTRA SCSI DRIVES				
Model	Description	EL	Int.	SR 2000
APS Q 2000	Quantum Fireball ST, 2079MB, 5.4K	\$199 ⁹⁵	\$219 ⁹⁵	\$289 ⁹⁵
APS Q 3000	Quantum Fireball ST, 3118MB, 5.4K	\$229 ⁹⁵	\$249 ⁹⁵	\$319 ⁹⁵
APS Q 4000	Quantum Fireball ST, 4136MB, 5.4K	\$279 ⁹⁵	\$299 ⁹⁵	\$369 ⁹⁵
APS Q 6400	Quantum Fireball ST, 6236MB, 5.4K	\$359 ⁹⁵	\$379 ⁹⁵	\$449 ⁹⁵
APS Q 4300	Quantum Atlas II, 4341MB, 7200 rpm		\$499 ⁹⁵	\$569 ⁹⁵
APS Q 9000	Quantum Atlas II, 8682MB, 7200 rpm		\$799 ⁹⁵	\$869 ⁹⁵
APS ST 2000	Seagate Barracuda, 2157MB, 7200 rpm		\$379 ⁹⁵	\$449 ⁹⁵
APS ST 4300	Seagate Barracuda, 4340MB, 7200 rpm		\$549 ⁹⁵	\$619 ⁹⁵
APS ST 4500	Seagate Cheetah, 4348MB, 10000 rpm		\$649 ⁹⁵	N/A
APS ST 9000	Seagate Barracuda, 8683MB, 7200 rpm		\$779 ⁹⁵	\$849 ⁹⁵
APS ST 9100	Seagate Cheetah, 8681MB, 10000 rpm		\$999 ⁹⁵	N/A
APS WD 2000	Western Digital Enterprise, 7200 rpm		\$379 ⁹⁵	\$449 ⁹⁵
APS WD 4300	Western Digital Enterprise, 7200 rpm		\$579 ⁹⁵	\$649 ⁹⁵

APS HIGH-PERFORMANCE ULTRA SCSI DRIVES		
Model	Description	Full Height
APS ST 23000	Seagate Elite 23, 22.1GB, 5400 rpm	\$1,799 ⁹⁵
APS ST 23000 W	Seagate Elite 23, 22.1GB, 5400 rpm	1,899 ⁹⁵

APS ULTRA WIDE SCSI DRIVES				
Model	Description	Int.	SR 2000	Pro
APS Q 4300 W	Quantum Atlas II, 4341MB, 7200 rpm	\$529 ⁹⁵	\$599 ⁹⁵	\$629 ⁹⁵
APS Q 4500 W	Quantum Viking, 4345MB, 7200 rpm	\$579 ⁹⁵	\$649 ⁹⁵	\$679 ⁹⁵
APS Q 9000 W	Quantum Atlas II, 8682MB, 7200 rpm	\$829 ⁹⁵	\$899 ⁹⁵	\$929 ⁹⁵
APS ST 2000 W	Seagate Barracuda, 2157MB, 7200 rpm	\$399 ⁹⁵	\$469 ⁹⁵	\$499 ⁹⁵
APS ST 4300 W	Seagate Barracuda, 4340MB, 7200 rpm	\$579 ⁹⁵	\$649 ⁹⁵	\$679 ⁹⁵
APS ST 4500 W	Seagate Cheetah, 4348MB, 10,000 rpm	\$699 ⁹⁵	N/A	\$799 ⁹⁵
APS ST 9000 W	Seagate Barracuda, 8683MB, 7200 rpm	\$799 ⁹⁵	\$869 ⁹⁵	\$899 ⁹⁵
APS ST 9100 W	Seagate Cheetah, 8681MB, 10,000 rpm	\$1,049 ⁹⁵	N/A	\$1,149 ⁹⁵
APS WD 2000 W	Western Digital Enterprise, 7200 rpm	\$429 ⁹⁵	\$499 ⁹⁵	\$529 ⁹⁵
APS WD 4300 W	Western Digital Enterprise, 7200 rpm	\$599 ⁹⁵	\$669 ⁹⁵	\$699 ⁹⁵

APS IDE DRIVES		
Model	Description	Internal
APS Q 3000	Quantum Fireball ST, 3118MB, 5400 rpm	\$199 ⁹⁵
APS Q 4000	Quantum Fireball ST, 4136MB, 5400 rpm	\$249 ⁹⁵
APS Q 6400	Quantum Fireball ST, 6236MB, 5400 rpm	\$329 ⁹⁵

APS POWERBOOK STORAGE		
Model	Description	Internal
APS T 2000	Toshiba MK2103MAV, 2067MB, 4200 rpm IDE	\$299 ⁹⁵

APS REMOVABLE DRIVES			
Model	Description	SR 1000	SR 2000
APS SQ 5200	SyQuest 5200, 190MB	N/A	\$389 ⁹⁵
APS Jaz	(with 1 cartridge) 1GB	\$399 ⁹⁵	\$449 ⁹⁵
APS Jaz II	(with 1 cartridge) 2GB	\$649 ⁹⁵	\$699 ⁹⁵

APS MO DRIVES			
Model	Description	SR 1000	SR 2000
APS 230 MO	(with 1 cartridge) 217MB	\$299 ⁹⁵	\$369 ⁹⁵
APS 640 MO	Fujitsu M2513A2#N	N/A	\$419 ⁹⁵
APS 2.6GB MO	Sony SMO-F544, 2.4GB	N/A	\$1,669 ⁹⁵

APS CD-ROM DRIVES			
Model	Description	External	
APS CD32	32X CD-ROM in Slimline Case	\$179 ⁹⁵	
APS CD Changer	Nakamichi 5 Disc/16X CD-ROM, Pro Case only	\$299 ⁹⁵	
APS CD-R Plus	2X record/6X read CD-R (Sony)	\$379 ⁹⁵	
APS CD-R Pro	4X record/6X read CD-R in Pro Enclosure	\$529 ⁹⁵	
APS CD-R Pro	4X record/6X read CD-R in Full Height Enclosure	\$729 ⁹⁵	
APS CD-RW	2X record/6X read CD-RW in Pro Enclosure	\$529 ⁹⁵	
APS CD-RW Pro	4X record/6X read/2X rewrite CD-RW, Pro Enclosure	\$599 ⁹⁵	
APS Jaz/CD-R System	2X record/6X read CD-R	\$799 ⁹⁵	
APS 2GB/CD-RW Sys.	Q 2000 Hard Drive & 2X record/6X read CD-RW	\$899 ⁹⁵	

*Available in an SR 2000 Enclosure for an additional \$50.00

APS TAPE BACKUP SYSTEMS			
Model	Description	Internal	External
APS HyperQIC®	Travan 4 Conner QIC 3095, 8GB	\$349 ⁹⁵	\$399 ⁹⁵
APS HyperDAT®	DDS-20C, 8GB	\$549 ⁹⁵	\$599 ⁹⁵
APS HyperDAT® Pro	DDS-20C, 8GB	\$849 ⁹⁵	\$899 ⁹⁵
APS HyperDAT® III	DDS-30C, 24GB	\$1,149 ⁹⁵	\$1,199 ⁹⁵
APS Mini Library	4mm DDS-2 AutoLoader, 64GB	N/A	\$2,399 ⁹⁵
APS DLT30	DLT 3000, 30GB	N/A	\$2,499 ⁹⁵
APS DLT40	DLT 4000, 40GB	N/A	\$2,999 ⁹⁵
APS DLT70	DLT 7000, 70GB	N/A	\$6,999 ⁹⁵
APS DLT*	8mm W/DC, 50GB	N/A	\$3,199 ⁹⁵
APS AIT*	4MM DDS-2 AutoLoader, 136GB	N/A	\$4,999 ⁹⁵

*See Adaptec PowerDomain 2940 Ultra Wide SCSI Host Adapter offer to gain maximum throughput. Call for information about Windows 95 and Windows NT versions of these great drive products

APS CD 32

- 32X for fluid multimedia!
- Fastest CD-ROM drive ever from APS

*Slimline case does not include DLT, but does come with block terminator.

APS CD-RW Pro \$599⁹⁵

- Work faster with 4x record, 2x rewrite & 6x read speeds
- Fastest rewritable CD technology available!

\$499⁹⁵ APS Q 4300

- Ideal for high-performance corporate applications
- 7200 rpm Ultra SCSI mechanism

Shown in SR 2000 Enclosure

Call 800-761-8133

Corporate Sales (800) 554-3941

International (816) 920-4109

APS Technologies • 6131 Deramus • Kansas City, MO 64120

Order from Our Online Catalog @

<http://www.apstech.com/store>

Check order status @ <http://www.apstech.com>

VISA MasterCard American Express Discover

IT'S BIGGER, TOUGHER, FASTER AND IT'S OUT.



CodeWarrior Professional Release 2. Just out and ready for some serious, industrial-strength programming. This is the only Integrated Development Environment that allows you to edit, compile and debug C, C++, Java and Pascal programs for multiple target processors and operating systems. We're still the codewriter's best friend and now we're better than ever. Here's why:

- ✕ portable project files — may be used interchangeably between MAC and Windows platforms
- ✕ file compare and merge — select two files to graphically view differences and merge changes
- ✕ browse across subprojects — source code browser works across targets and subprojects

You can count on CodeWarrior Professional Release 2. We're powerful, proven, easy-to-handle and we've had all our shots.

CodeWarrior
mark your territory.



metrowerks®

www.metrowerks.com